Latent Dirichlet Allocation, explained and improved upon for applications in marketing intelligence

Iris Koks

Latent Dirichlet Allocation, explained and improved upon for applications in marketing intelligence

by

Iris Koks

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Friday March 22, 2019 at 2:00 PM.

Student number:	4299981	
Project duration:	August, 2018 – March, 2019	
Thesis committee:	Prof. dr. ir. Geurt Jongbloed,	TU Delft, supervisor
	Dr. Dorota Kurowicka,	TU Delft
	Drs. Jan Willem Bikker PDEng,	CQM

Consultants in Quantitative Methods

"La science, mon garçon, est faite d'erreurs, mais ce sont des erreurs qu'il est utile de faire, parce qu'elles conduisent peu à peu à la vérité."

- Jules Verne, in "Voyage au centre de la Terre"

Abstract

In today's digital world, customers give their opinions on a product that they have purchased online in the form of reviews. The industry is interested in these reviews, and wants to know about which topics their clients write, such that the producers can improve products on specific aspects. Topic models can extract the main topics from large data sets such as the review data. One of these is Latent Dirichlet Allocation (LDA). LDA is a hierarchical Bayesian topic model that retrieves topics from text data sets in an unsupervised manner. The method assumes that a topic is assigned to each word in a document (review), and aims to retrieve the topic distribution for each document, and a word distribution for each topic. Using the highest probability words from each topic-word distribution, the content of each topic can be determined, such that the main subjects can be derived. Three methods of inference to obtain the topic and word distributions are considered in this research: Gibbs sampling, Variational methods, and Adam optimization to find the posterior mode. Gibbs sampling and Adam optimization have the best theoretical foundations for their application to LDA. From results on artificial and real data sets, it is concluded that Gibbs sampling has the best performance in terms of robustness and perplexity.

In case the data set consists of reviews, it is desired to extract the sentiment (positive, neutral, negative) from the documents, in addition to the topics. Therefore, an extension to LDA that uses sentiment words and sentence structure as additional input is proposed: LDA with syntax and sentiment. In this model, a topic distribution and a sentiment distribution for each review are retrieved. Furthermore, a word distribution per topic-sentiment combination can be estimated. With these distributions, the main topics and sentiments in a data set can be determined. Adam optimization is used as inference method. The algorithm is tested on simulated data and found to work well. However, the optimization method is very sensitive to hyperparameter settings, so it is expected that Gibbs sampling as inference method for LDA with syntax and sentiment performs better. Its implementation is left for further research.

Keywords: Latent Dirichlet Allocation, topic modeling, sentiment analysis, opinion mining, review analysis, Hierarchical Bayesian inference

Preface

With this thesis, my student life comes to an end. Although I started with studying French after high school, after 4 years, I finally came to my senses, such that now, I have become a mathematician. Fortunately, my passion for languages has never completely disappeared, and it could even be incorporated in this final research project.

During the last 8 months, I have been doing research and writing my master thesis at CQM in Eindhoven. I have had a wonderful time with my colleagues there, and I would like to thank all of them for making it the interesting, nice time it was. I learned a lot about industrial statistics, machine learning, consultancy and the CQM-way of working. Special thanks go to my supervisors Jan Willem Bikker, Peter Stehouwer and Matthijs Tijink, for their sincere interest and helpfulness during every meeting. Next to the interesting conversations about mathematics, we also had nice talks about careers, life and personal development, which I will always remember.
Although he was not my direct supervisor, Johan van Rooij helped me a lot with programming and implementation questions, and by thinking along patiently when I got stuck in some mathematical derivation, for which I am very grateful.

Furthermore, I would like to thank my other supervisor, professor Geurt Jongbloed, for his guidance through this project and for bringing out the best of me on a mathematical level. Naturally, I also enjoyed our talks about all aspects of life and the laughs we had. Also, I would like to thank Dorota Kurowicka for being on my graduation committee, and for her interesting lectures about copulas. Unfortunately, I could not incorporate them in this project, but maybe in my career as (financial) risk analyst?

Then, of course, I would like to express my gratitude to Jan Frouws, for his unconditional care and support during these last 8 months. He was always there to listen to me, mumbling on and on about reviews, topics, strollers and optimization methods. Without you, throughout this project, the ups would not have been that high and the downs would have been way deeper. I am really proud of how we manage and enjoy life together.

In addition, I would like to thank my parents for always supporting me during my entire educational journey. Because of them, I always liked going to school and learning, hence my broad interests in languages, economics, physics and mathematics. Also, thanks to my brother Corné, with whom I made my homework together for years and I had the most interesting (political) discussions.

Lastly, I would like to thank my friends that I met in all different places. With you, I have had a wonderful student life in which I started (and immediately quit) rowing, bouldering, learned to cook properly and for large groups (making sushi, pasta, ravioli, mexican wraps, massive Heel Holland Bakt cakes...), learned about Christianity, air planes and bit coins, lost my 'Brabants' accent such that I became a little more like a 'Randstad' person, and gained interest in the financial world, in which I hope to get a nice career.

Iris Koks
Eindhoven, March 2019

Nomenclature

$\mathbb{T}_n$	n-dimensional closed simplex
$\mathscr{D}$	Corpus, set of all documents
$\mathscr{L}$	Log likelihood
Σ	Number of different sentiments
σ	Sentiment index
$\boldsymbol{\alpha}$	Hyperparameter vector of size K with belief on document-topic distribution
$\boldsymbol{\beta}$	Hyperparameter vector of size V with belief on topic-word distribution
$\boldsymbol{\gamma}$	Hyperparameter vector of size Σ with belief on document-sentiment distribution
$\boldsymbol{\phi}_\mathbf{k}$	Word probability vector of size V for topic k
$\boldsymbol{\pi}_\mathbf{d}$	Sentiment probability vector of size Σ for document d
$\boldsymbol{\theta}_\mathbf{d}$	Topic probability vector of size K for document d
C	Number of different parts-of-speech considered
c	Part-of-speech
d	Document index
H	Entropy
h	Shannon information
K	Number of topics
M	Number of documents in a data set
N_d	Number of words in document d
N_s	Number of words in phrase s
s	Phrase or sentence index
S_d	Number of phrases in document d
V	Vocabulary size, i.e. number of unique words in data set
w	Word index
z	Topic index
Adam	Adaptive moment estimation (optimization method)
JS	Jensen-Shannon
KL	Kullback-Leibler
LDA	Latent Dirichlet Allocation
MAP	Maximum a posteriori or posterior mode
NLP	Natural Language Processing
VBEM	Variational Bayesian Expectation Maximization

Contents

List of Figures

List of Tables

1 Introduction

The last decade, giant steps have been made in the world of big data and 'big analytics'. These terms are used in several settings, and their definitions evolve; what is now called 'big' data will probably not be that big anymore in a few years. With the availability of big data and fast computers, better and large-scale analyses can be done. For companies, these analyses are key, as it is believed that lots of information and knowledge can be retrieved from the logged data and data that is freely available online. The field of applications of big data that this thesis focuses on is marketing intelligence. That is, using data to gain insights into customer behavior and opinions. Even marketing strategies can be tuned based on prediction models such that the strategy is optimal for sales or product ratings. In the next section, we will dive into the specific questions CQM[1] is asked by their clients.

1.1. Costumer insights using Latent Dirichlet Allocation

Consider you are head of marketing of a large industrial company. On the box of your product is a claim, for example, *'easy to use and unbreakable'*. It is expected that this text motivates the customer to buy your product. You are interested in the influence of this claim on customer opinion, so we resort to online reviews. Do people talk online about the claim on the box? If there are also boxes with different claims for the same product, is there a significant difference in opinion when it comes to, e.g. ease of use? The answers to these questions can help the marketing department choosing the text on the box well and gain more knowledge on the customer experience. This information will help the company with better meeting the customers' needs and wishes.
Another question concerns the star rating of a product on a webshop. These star ratings are essential to industrial companies because they are sometimes even linked to (personal) bonuses. That is why these companies desire to know what aspects of the product drive the star rating. Are customers more satisfied if the product is cheap but satisfactory, or is instead the ease of use more critical in their final judgment, and thus the star rating?

Naturally, it is very time-consuming to read all reviews online, so we want to apply a method that quickly summarizes a (large) set of reviews in a list of topics. This is where the field of topic modeling comes in. A topic model is a statistical method that retrieves topics from a set of documents, consisting of words. Here topics should be seen as common themes or subjects that occur in many documents. For the review data, we can think about for example the price; a large group of customers is expected to mention the price of a product in their review.
One of the simplest topic models is Latent Dirichlet Allocation (LDA). This model assumes that there is a set number of topics in the set of documents/reviews, and finds distributions over the topics for each document and distributions over a list of words for each topic. With these distributions, we know in what proportions customers write about specific themes, and per topic, we know what are the most frequently used words. From these lists of highly probable words per topic, the general customer opinion on that topic can usually be

[1] CQM stands for Consultants in Quantitative Methods and is the company in which my internship took place. The department of CQM in which this thesis is written is specialized in product and process innovation. To better innovate and improve products, insights in customer opinions are required. Therefore, research is done in extracting overarching themes and opinions from large sets of online reviews, without having to read every single review.

concluded. Latent Dirichlet Allocation is therefore well applicable for review analyses and forms the main theme of this thesis.

1.2. Research questions

This research is conducted in collaboration with CQM. Therefore, the research questions that are studied in this thesis, follow from issues encountered in the day-to-day work at CQM.
First of all, the Bayesian model called Latent Dirichlet Allocation needs a more thorough explanation. There are many different applications of LDA in software, but a mathematical explanation of the method of inference used in that software often lacks. Also in literature, articles might offer too little information on what is really behind the model of Latent Dirichlet Allocation, and different methods of inference are proposed. An overview of these methods including extensive derivations is thus given in this thesis. Secondly, Latent Dirichlet Allocation is a model that can be applied to all kinds of documents. However, in the field of marketing intelligence, LDA might not give the desired results, and more information is expected to be 'hidden' in the analyzed reviews. An improvement upon the basic LDA model to make it more suitable for review analyses is researched.

As a conclusion, two research questions that are to be answered in this thesis.

- What is Latent Dirichlet Allocation and which methods of inference are possible for this Bayesian hierarchical model?
- How can Latent Dirichlet Allocation be improved upon to make it more suitable for review analyses using linguistics?

1.3. Thesis outline

At the moment, you, the reader, have already been given the motivation to use Latent Dirichlet Allocation. Before being able to elaborate on the precise working principle of LDA model in chapter 3, in chapter 2 some theoretical background information essential to understanding chapter 3 is given. Apart from fundamentally understanding what LDA does and what assumptions are made, different methods of inference are possible and are explained in chapter 4. To make a clear distinction between which methods are already frequently used in the literature, and which method is a new contribution of this thesis, in chapter 4 only the inference methods that are present in literature and widely used are explained. In chapter 5, a new, different inference method is explained, which is, to the best of author's knowledge, not yet applied to LDA.
Because CQM gets specific questions from clients about customer opinions on products, an extension of LDA that is, in particular, suitable to extract this kind of information from large data sets is described in chapter 6. This extension has been given the name 'LDA with syntax and sentiment', as it extends plain LDA with a combination of the sentiment of words (i.e., positive and negative opinion words) with the their part-of-speech. Then, in chapter 7, a small note is made on how to interpret the results of LDA and which conclusions should and should not be drawn. In chapter 8, the actual results of the application of LDA to different data sets are displayed. Firstly, the different inference methods are compared, and secondly, results for LDA with syntax and sentiment are given. Naturally, a discussion on the results is present in chapter 9, and lastly, in chapter 10, conclusions on the different inference methods for LDA, and the extended version of LDA specifically designed for CQM are drawn. Also, recommendations on further research can be read about in chapter 10.

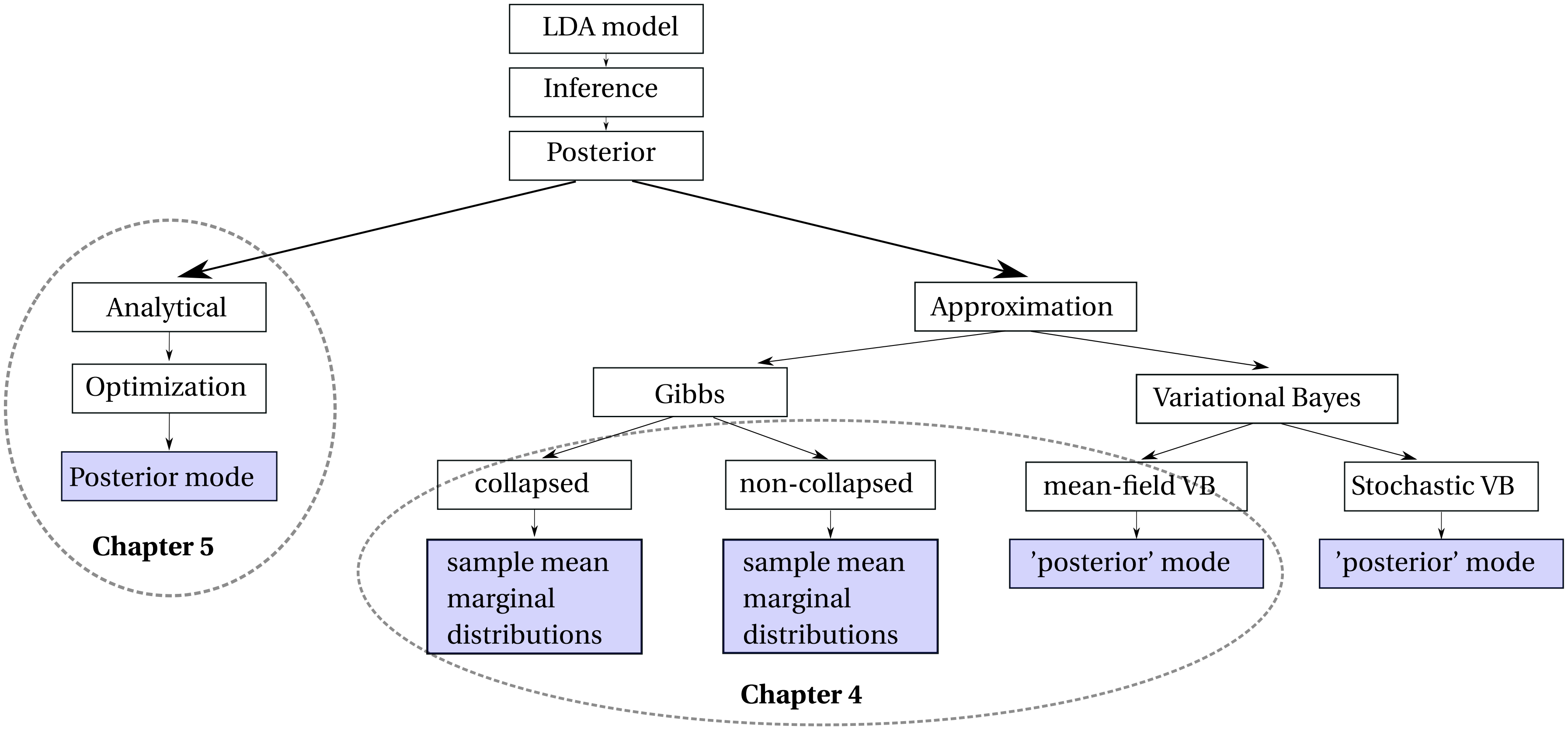


Figure 1.1: Overview of methods of inference to estimate model parameters of Latent Dirichlet Allocation. The posterior distribution can either be calculated analytically, after which the posterior model can be found through optimization, or it can be approximated using Gibbs sampling or Variational Bayes methods. The latter methods are discussed in chapter 4 of this thesis, while the first inference method is described in chapter 5, as it is one of the main contributions of this research.

1.4. Note on notation

In the field of mathematics, there are many different ways of saying the same. Therefore, in this section, some clarity is given about the notations used in this thesis.

First, all constants or one-dimensional parameters are simply given by its letter in italic, albeit from the Greek or Roman alphabet, or upper or lower case. Then, when the parameter is a vector, the corresponding letter is given in boldface. Sometimes, for ease of notation, also sets of vectors are given in boldface, such that $\boldsymbol{\phi} = \{\boldsymbol{\phi_1}, \dots, \boldsymbol{\phi_K}\}$, while, strictly speaking, $\boldsymbol{\phi}$ is a set of vectors. In case this simplified notation is used, it will have been mentioned.
Very often in this thesis, only one element of a vector is used in an equation or a density. This element is then denoted with a subscript, while the vector remains in boldface and is surrounded with round brackets. For example, the ith element of vector $\boldsymbol{\phi_j}$ is denoted with $(\boldsymbol{\phi_j})_i$ (where j is used to indicate which vector $\boldsymbol{\phi}$ is used, as there are many vectors $\boldsymbol{\phi}$).

Secondly, random variables are, conventionally, denoted with capital letters. Once they take a value, the notation changes to lower case to show that we are dealing with data. There are some constants also denoted with a capital letter, such as the data set size. When in doubt, the nomenclature can be consulted for clarification. As conventional, the expectation of a random variable is given by $\mathbb{E}$. If a probability of a random variable X taking value x, is denoted with $\mathbb{P}(X = x)$, we use $\mathbb{P}$ to indicate the probability measure that belongs to the probability space in which X lives.

Lastly, for densities, we use the Bayesian notation, as will be explained in chapter 2 more extensively. With the Bayesian notation, we mean that the density of random variable X, $f_X(x)$, is denoted with $p(x)$. The conditional density of random variable X given Y, $f_{X|Y}(x|y)$, becomes $p(x|y)$.

2 Theoretical background

"Inside every non-Bayesian, there is a Bayesian struggling to get out"
Dennis Lindley (1923-2013)

This chapter contains the theoretical background needed to understand the rest of this thesis. Firstly, the principles of Bayesian statistics are explained, since LDA is a hierarchical Bayesian model. Secondly, a section is dedicated to the Dirichlet process and Dirichlet distribution, because the latter is used multiple times in the model.
Next, natural language processing (NLP), an overlapping field in computer science, artificial intelligence and linguistics, is introduced. Some principles of NLP are used in the preprocessing steps of review analyses with LDA. Lastly, model selection criteria that are frequently used in topic modeling are explained.

2.1. Bayesian statistics

In the field of statistics, there are two types of statisticians (generally speaking): frequentists and Bayesian believers. The latter group considers all unknowns to be random variables, including the parameters [46]. We will explain this way of thinking and doing statistics using a simple example.

Consider flipping a coin of which we do not know whether it is fair or not. Suppose we do this n times, and $X_i|\Theta \sim \text{Bernoulli}(\Theta)$ for $i = 1,\dots,n$. The probability of throwing heads is represented by **random variable** Θ, while the probability of getting tails is $1-\Theta$. The random variables $X_1,\dots,X_n$ are the flips and can either be heads or tails, i.e. $X_i \in \{H,T\}$, for $i = 1,\dots,n$. Note that each flip is executed separately and with the same coin (and other circumstances), therefore $X_1,\dots X_n$ are **conditionally independent** and identically distributed. Note that conditional independence is specific for Bayesian statistics, as parameter Θ is a random variable. Therefore, conditional on Θ, the flips are independent. Because Θ is a random variable, it has a distribution reflecting our belief.

Beforehand, we believe that the probability of heads is in the neighborhood of $\frac{1}{2}$, as would be the case for a fair coin. This belief is inserted into the model via a prior distribution. This is the distribution of Θ we believe to be true before generating the data. After having observed $X_1 = x_1,\dots,X_n = x_n$, the posterior distribution is constructed, as the name suggests. The posterior density is determined via Bayes' rule:

$$f_{\Theta|\mathbf{X}}(\theta|\mathbf{x}) = \frac{f_{\mathbf{X}|\Theta}(\mathbf{x}|\theta)\cdot f_{\Theta}(\theta)}{f_{\mathbf{X}}(\mathbf{x})} \tag{2.1}$$

Here $f_{\Theta|\mathbf{X}}(\theta|\mathbf{x})$ is the conditional density of parameter Θ given the observations $\mathbf{X} = \mathbf{x}$, and is referred to as the posterior. $f_{\mathbf{X}|\Theta}(\mathbf{x}|\theta)$ is the conditional multivariate density function of random variables $X_1,\dots X_n$ given parameter $\Theta = \theta$, $f_{\Theta}(\theta)$ is the density of the initial distribution of Θ, that is the prior density, and the term in the denominator, $f_{\mathbf{X}}(\mathbf{x})$, is called the evidence. This is the marginal distribution of the data and can be determined by integrating the numerator in equation 2.1 over θ.

An appropriate prior that represents our strong belief that the coin will have approximately equal probabilities for heads and tails, is given by the Beta distribution with equal parameters and thus mean $\frac{1}{2}$. Because we are relatively certain that the coin is fair, we choose both a and b to be 4. Therefore, we take as prior $\Theta \sim \text{Beta}(4,4)$:

$$f_\Theta(\theta) = \frac{1}{B(4,4)}\theta^3 \cdot (1-\theta)^3 \tag{2.2}$$

With $\mathrm{B}(a,b)$ the beta function, defined as:

$$\mathrm{B}(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}\,dt \tag{2.3}$$

Which can be rewritten in terms of the gamma function $\Gamma(a) = \int_0^\infty t^{a-1}e^{-t}\,dt$, as shown in e.g. [37]:

$$\mathrm{B}(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} \tag{2.4}$$

Instead of determining $f_{\mathbf{X}|\Theta}(\mathbf{x}|\theta)$ directly, it is wise to define Y as the total number of heads first, such that $Y \sim \text{Binomial}(n,\Theta)$. The probability density function of Y is then:

$$f_{Y|\Theta}(y|\theta) = \frac{n!}{y!(n-y)!}\theta^y \cdot (1-\theta)^{n-y} \tag{2.5}$$

Using Bayes' rule from equation 2.1, the posterior distribution becomes:

$$\begin{aligned} f_{\Theta|\mathbf{X}}(\theta|x_1,\dots,x_n) &= f_{\Theta|Y}\left(\theta|Y = \sum_{i=1}^n \mathbb{1}_{x_i=H}\right) && (*) \\ &\propto f_{Y|\Theta}(y|\theta)\cdot f_\Theta(\theta) && (**) \\ &= \frac{n!}{y!(n-y)!}\theta^y\cdot(1-\theta)^{n-y}\cdot\frac{1}{B(4,4)}\theta^3\cdot(1-\theta)^3 \\ &\propto \theta^{y+3}\cdot(1-\theta)^{n-y+3} \end{aligned} \tag{2.6}$$

(*): Because the data $x_1,\dots,x_n$ is categorical data, namely heads (H) or tails (T), it is better to work with random variable Y, as introduced above.
(**): We only take the part of equation 2.1 that depends on θ because we are merely interested in the distribution of this random variable. The denominator of 2.1 does not depend on θ, therefore this term is left out.
Note that both n and y are known; n is fixed, and y is given by the data. In equation 2.6, θ is the only unknown. We recognize in the posterior density in equation 2.6 a Beta density with parameters $y+4$ and $n-y+4$, i.e., $\Theta|\mathbf{X} \sim \text{Beta}(y+4, n-y+4)$.

The posterior distribution does not give us directly the value of Θ, it is only a density over all values that Θ can attain. A natural way to get an estimator from the posterior is to compute the mean or the mode of the distribution. These can be straightforwardly determined using:

$$\text{posterior mean} = \hat{\Theta}_{\text{mean}} = \int_0^1 \theta\cdot f_{\Theta|\mathbf{X}}(\theta|\mathbf{x})d\theta \tag{2.7}$$

$$\text{posterior mode} = \hat{\Theta}_{\text{mode}} = \arg\max_{\theta'}\left\{f_{\Theta|\mathbf{X}}(\theta'|\mathbf{x})\right\} \tag{2.8}$$

For the beta distribution, these two estimators can easily be computed, as their expressions are known (and easily derived) in terms of the parameters. The two estimators for our example with flipping a coin n times become:

$$\hat{\Theta}_{\text{mean}} = \frac{y+4}{n+8} \tag{2.9}$$

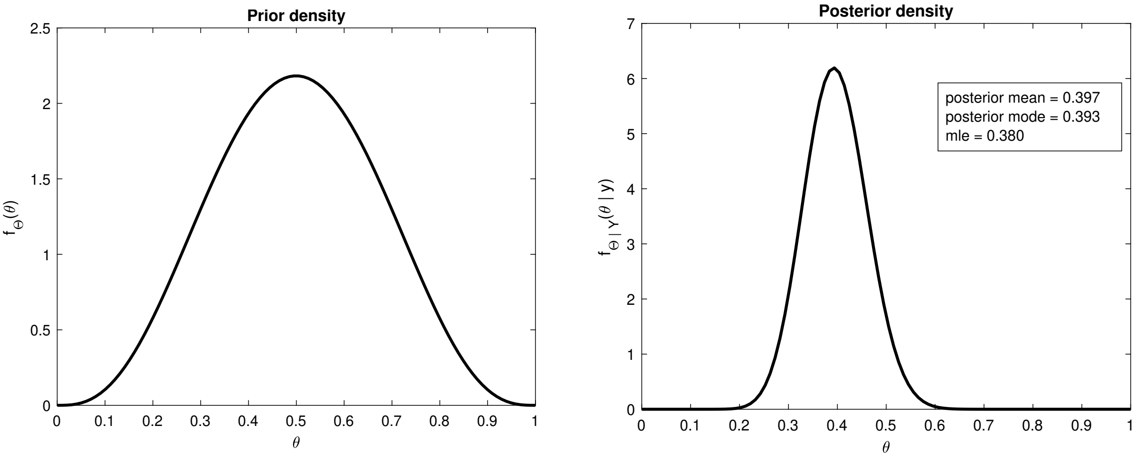
$$\hat{\Theta}_{\text{mode}} = \frac{y+3}{n+6} \tag{2.10}$$

When considering the coin flipping experiment as a frequentist statistician, a possible estimator for the probability of heads, θ would be the maximum likelihood estimator. It is easy to verify that:

$$\hat{\theta}_{\mathrm{MLE}} = \frac{y}{n} \tag{2.11}$$

Comparing this estimator with the posterior mean and mode above, we can observe the influence of the prior distribution. Also note that for increasing sample size n, the influence of the prior diminishes. This principle is generalized in the Bernstein-Von Mises theorem in e.g. [46].

The shift of the prior density of Θ to the posterior density $f_{\Theta|\mathbf{X}}(\theta|\mathbf{x})$ is visualized in figure 2.1. The used data is drawn from a Bernoulli(0.3) distribution, so the true parameter is $\theta = 0.3$. The sample size is $n = 50$, and we can see that the three estimators, the posterior mean and mode, and the maximum likelihood estimator, are still quite far off with respect to the true θ. For the maximum likelihood estimator, this is caused by the fact that the sample size is not large enough for a precise estimator. It is clear that if the sample size is increased, the $\hat{\theta}_{MLE}$ will be closer to 0.3. The posterior mean and mode are influenced by the prior distribution that attributes most mass to $\theta = 0.5$. That is the reason why the values of these estimators are closer to 0.5 than to the actual value 0.3.



(a) prior: $\Theta \sim \mathrm{Beta}(4,4)$

(b) posterior: $\Theta \sim \mathrm{Beta}(y+4, n-y+4)$

Figure 2.1: Bayesian statistics: A prior density is imposed on the random parameter Θ and results in a posterior density of Θ given the observed data. The three possible estimators for Θ are given in figure 2.1b.

Latent Dirichlet Allocation is slightly more complicated than the example above. In the example, we used the fact that a Beta distribution is conjugate to the binomial distribution. That is, if we choose a Beta distribution as a prior, and the data is binomially distributed given the random parameter, the posterior distribution will also be Beta distributed. In Bayesian statistics, these conjugate priors are often used to simply inference. In LDA, instead of Beta and Binomial distributions, we use Dirichlet and Multinomial distributions, which are multivariate versions of the former. More on these distributions and their application in LDA will be explained in chapter 3.

A last note needs to be made on the notation of the prior and posterior densities in Bayesian statistics. Instead of writing $f_{\Theta|\mathbf{X}}(\theta|\mathbf{x})$, usually the shorter version $p(\theta|\mathbf{x})$ is used. Similarly, $p(\theta)$ denotes the prior density and $p(\mathbf{x}|\theta)$ the likelihood. This 'Bayesian' notation will be used in the remainder of this thesis.

2.2. Dirichlet process and distribution

Latent Dirichlet Allocation uses, as the name suggests, a Dirichlet distribution as a prior twice; once for the document-topic distributions, and once for the topic-word distributions. A Dirichlet distribution can be thought of as a distribution over distributions. Where the latter distribution always consists of a probability vector, that is a vector of which each element represents a probability, and of which the elements sum to one. The official definition and properties of the Dirichlet distribution are given in subsection 2.2.2, but we first focus on the representation of sampling from a Dirichlet distribution using the stick-breaking construction. This method helps the reader understand the underlying principles that characterize the Dirichlet distribution.

2.2.1. Stick-breaking construction of Dirichlet process

A common representation of sampling from a Dirichlet distribution is given by the stick-breaking construction, introduced by Sethuraman in [40]. We will first state the constructive definition of a Dirichlet distribution, after which the intuition behind it will be explained.
If a vector $\boldsymbol{\theta}$ of length K is constructed according to:

$$\begin{aligned}\boldsymbol{\theta} &= \sum_{i=1}^{\infty} V_i \prod_{j=1}^{i-1}(1-V_j)\mathbf{e}_{Y_i} \\ V_i &\sim \text{Beta}(1,\alpha) \qquad i.i.d. \\ Y_i &\sim \text{Multinomial}(1,\mathbf{g_0}) \qquad i.i.d. \qquad\qquad (*),\end{aligned} \tag{2.12}$$

then $\boldsymbol{\theta} \sim \text{Dirichlet}(\alpha\cdot\mathbf{g_0})$. At $(*)$, a draw from a Multinomial$(1,\mathbf{g_0})$ results in a unit vector, with in one dimension $i \in \{1,\dots,K\}$ all probability mass, and zero probability mass in all other dimensions. Y_i is then the index of the dimension in which all mass is concentrated. The vector $\mathbf{e}_{Y_i}$ is the unit vector in the dimension Y_i, as $Y_i \in \{1,\dots,K\}$. As conventionally, i.i.d. means independent and identically distributed.
For the proof of $\boldsymbol{\theta}$ from equation 2.12 having the same distribution as $\boldsymbol{\theta} \sim \text{Dirichlet}(\alpha\cdot\mathbf{g_0})$, we refer to [34].

Let us go step by step through the process. For $i = 1$, we draw a V_1 from the Beta distribution and a Y_1 from the Multinomial distribution. Y_1 denotes a dimension, and V_1 denotes the mass assigned to that dimension in the vector $\boldsymbol{\theta}$. Then, for the second iteration, again a dimension Y_2 and a length V_2 are drawn. The mass V_2 is assigned to dimension Y_2 in $\boldsymbol{\theta}$, while mass $V_2\cdot(1-V_1)$ is assigned to the initially drawn dimension Y_1 of $\boldsymbol{\theta}$. Note that Y_1 might be the same as Y_2, as they are drawn from the same distribution and they are independent random variables.
It is easy to see that the probability mass for each i, $V_i\prod_{j=1}^{i-1}(1-V_j)$, lies between 0 and 1. This means that we can look at the assignment of the mass in each iteration as consequently breaking a stick, hence the name. We start with a stick of unit length and break V_1 from it. Then, from the remainder of the stick, V_2 is broken. This continues for $i \to \infty$, such that in total, the mass is distributed over the K dimensions as $\boldsymbol{\theta}$.
Parameters α and $\mathbf{g_0}$ influence this distribution. The smaller α, the larger the probability of drawing a large V_i, such that only in the first few iterations, almost all mass is already distributed. On the other hand, if α is large, the density of the Beta$(1,\alpha)$ will be skewed to the left, and the V_i will be small, resulting in the end in a more uniformly distributed probability vector $\boldsymbol{\theta}$, given a symmetric $\mathbf{g_0}$. This second parameter, $\mathbf{g_0}$, handles the preference to certain dimensions. If $(\mathbf{g_0})_1$ is much larger than all other elements of $\mathbf{g_0}$, most mass will be assigned to the first dimension in the iterative process of equations 2.12, such that $(\boldsymbol{\theta})_1$ will be much larger than all other elements of $\boldsymbol{\theta}$.
It can be concluded that α is a scaling parameter, which with can be steered towards a more uniform distribution, or, on the other hand, a distribution that assigns most probability mass to one or a few dimensions. With $\mathbf{g_0}$, we can incorporate preference to certain dimensions in the distribution. It can be seen as a location parameter. If $\mathbf{g_0}$ is symmetric and consists of, for example, only ones, Y_i can take on each dimension with equal probability in every step i in the constructive process.

In general, the parameter vector of a Dirichlet distribution is given by $\boldsymbol{\alpha}$, in which both the scaling as the location parameter are collected. In the next section, the general definition of a Dirichlet distributed random variable will be given, and its properties are derived and visualized.

2.2.2. Dirichlet distribution

Now that the process of sampling from a Dirichlet distribution is explained, we have gained understanding of the parameters of this distribution and their functions. For the official definition, we first need to understand the simplex. In this thesis, it is chosen to define the Dirichlet distribution using the closed simplex, as in [33].

Definition 2.1 (Closed simplex)
Let c be a positive number. The n-dimensional closed simplex $\mathbb{T}_n(c)$ in $\mathbb{R}^n$ is defined by[1]*:*

$$\mathbb{T}_n(c) = \left\{ (x_1, \dots, x_n)^T : x_i > 0,\ 1 \le i \le n,\ \sum_{i=1}^{n} x_i = c \right\}$$

An alternative is the open simplex, in which we define the sum of $\sum_{i=1}^{n-1} x_i$ to be smaller than constant c. It is a matter of choice to use either the open or closed simplex, as long as we ensure that the elements of a Dirichlet distributed random vector sum to 1 and live in $(0,1)^n$ for an n-dimensional vector:

Definition 2.2 (Dirichlet density [33])
A random vector $\mathbf{X} = (X_1, \dots, X_n)^T \in \mathbb{T}_n(1)$ is said to have a Dirichlet distribution if the density of $\mathbf{X}_{-n} = (X_1, \dots, X_{n-1})^T$ is:

$$\text{Dirichlet}_n(\mathbf{X}_{-n}|\boldsymbol{\alpha}) = \frac{\Gamma\left(\sum_{i=1}^{n} \alpha_i\right)}{\prod_{i=1}^{n} \Gamma(\alpha_i)} \prod_{i=1}^{n} x_i^{\alpha_i - 1} \tag{2.13}$$

where $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)^T$ is a strictly positive parameter vector. We will write $\mathbf{X} \sim \text{Dirichlet}_n(\boldsymbol{\alpha})$ on $\mathbb{T}_n(1)$. That is $\sum_{i=1}^{n} X_i = 1$.

The Dirichlet distribution thanks its name to the integral in 2.14, studied by Peter Gustav Lejeune Dirichlet in 1839, to which the integral of the Dirichlet density from equation 2.13 is proportional.

$$\int \left(\prod_{i=1}^{n-1} x_i^{\alpha_i - 1} \right) \left(1 - \sum_{i=1}^{n-1} x_i \right)^{a_n - 1} dx_1 \cdots dx_{n-1} = \frac{\prod_{i=1}^{n} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{n} \alpha_i\right)} \tag{2.14}$$

Furthermore, note that for $n = 2$, we obtain a Beta distribution.

$$\begin{aligned} \text{Beta}(\alpha_1, \alpha_2) &= \frac{\Gamma(\alpha_1 + \alpha_2)}{\Gamma(\alpha_1)\Gamma(\alpha_2)} x^{\alpha_1 - 1} (1-x)^{\alpha_2 - 1} \\ &= \frac{1}{B(\alpha_1, \alpha_2)} x^{\alpha_1 - 1} (1-x)^{\alpha_2 - 1} \end{aligned} \tag{2.15}$$

Consequently, the Dirichlet distribution can be thought of as a higher dimensional version of the Beta distribution.

Properties

The Dirichlet distribution has nice closed-form properties when it comes to marginal distributions, conditional distributions and product moment generating functions. Most of them are used in this thesis.

First, we take a look at the theorem containing the marginal and conditional distributions from [33]. The proof of this theorem can be found in [33].

Theorem 2.1 (Marginals and conditionals [33])
Let $\mathbf{X} \sim \text{Dirichlet}_n(\boldsymbol{\alpha})$ on $\mathbb{T}_n$, then we have the following results.

1. *For any $s < n$, the subvector $(X_1, \dots, X_s)^T$ has a Dirichlet distribution with parameters $(\alpha_1, \dots, \alpha_s; \sum_{j=s+1}^{n} \alpha_j)$. In particular, $X_i \sim \text{Beta}(\alpha_i, \alpha_+ - \alpha_i)$ with $\alpha_+ = \sum_{i=1}^{n} \alpha_i$.*
2. *The conditional distribution of $X_i' = \frac{X_i}{1 - \sum_{j=1}^{s} x_j^*}$ for $i \in \{s+1, \dots, n-1\}$ given $X_1 = x_1^*, \dots, X_s = x_s^*$, follows a Dirichlet distribution with parameters $(\alpha_{s+1}, \dots, \alpha_{n-1}, \alpha_n)$.*

To give an idea of the proof of the marginal distributions, we show the derivation for a one-dimensional marginal distribution of a three-dimensional Dirichlet distribution. The method used in this derivation, can be informative for other derivations in this thesis.
We want to show that the marginal distribution of a 3-dimensional Dirichlet distribution on $\mathbb{T}_n(1)$ is a Beta distribution with parameters α_i and $\sum_{j=1, i \neq j}^{3} \alpha_j$, following theorem 2.1. This result can later be generalized

[1] Note that this is not a closed space in the topological sense.

for a n-dimensional Dirichlet distribution. For ease of notation, let us derive the margin of X_1. Because $X_3 = 1 - X_1 - X_2$, we only need to integrate out X_2.

$$\begin{aligned}
f_{X_1}(x_1) &= \frac{\Gamma(\sum_{i=1}^{3}\alpha_i)}{\prod_{i=1}^{3}\Gamma(\alpha_i)}\int_0^1 x_1^{\alpha_1-1}x_2^{\alpha_2-1}(1-x_1-x_2)^{\alpha_3-1}\,dx_2\\
&= \frac{\Gamma(\sum_{i=1}^{3}\alpha_i)}{\prod_{i=1}^{3}\Gamma(\alpha_i)}x_1^{\alpha_1-1}\int_0^{1-x_1} x_2^{\alpha_2-1}(1-x_1-x_2)^{\alpha_3-1}\,dx_2 \qquad *\\
&= \frac{\Gamma(\sum_{i=1}^{3}\alpha_i)}{\prod_{i=1}^{3}\Gamma(\alpha_i)}x_1^{\alpha_1-1}\int_0^1 (1-x_1)^{\alpha_2-1}u^{\alpha_2-1}(1-x_1)^{\alpha_3-1}(1-u)^{\alpha_3-1}(1-x_1)\,du \qquad **\\
&= \frac{\Gamma(\sum_{i=1}^{3}\alpha_i)}{\prod_{i=1}^{3}\Gamma(\alpha_i)}x_1^{\alpha_1-1}(1-x_1)^{\alpha_2+\alpha_3-1}\int_0^1 u^{\alpha_2-1}(1-u)^{\alpha_3-1}\,du\\
&= \frac{\Gamma(\sum_{i=1}^{3}\alpha_i)}{\prod_{i=1}^{3}\Gamma(\alpha_i)}x_1^{\alpha_1-1}(1-x_1)^{\alpha_2+\alpha_3-1}\frac{\Gamma(\alpha_2)\Gamma(\alpha_3)}{\Gamma(\alpha_2+\alpha_3)}\\
&= \frac{\Gamma(\alpha_1+\alpha_2+\alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2+\alpha_3)}x_1^{\alpha_1-1}(1-x_1)^{\alpha_2+\alpha_3-1}
\end{aligned}\tag{2.16}$$

$*$ Combining $0 \leq x_2 \leq 1$ with $0 \leq 1 - x_1 - x_2 \leq 1$ results in $0 \leq x_2 \leq 1 - x_1$.
$**$ Substitution of $x_2 = (1 - x_1)u$.

The last expression in 2.16 is exactly the density function of a Beta(α_1,$\alpha_2 + \alpha_3$) distributed random variable.

This result can be generalized for an n-dimensional Dirichlet distribution by integrating out all variables x_j for $j \neq i, j \in \{1,\dots,n-1\}$. Note that the Dirichlet distribution of an n-dimensional vector has support on the $(n-1)$-simplex by definition. For x_n we therefore use $1 - x_1 - \cdots - x_{n-1}$. The same result follows:

$$X_i \sim \text{Beta}\left(\alpha_i, \sum_{j=1, j\neq i}^{n}\alpha_j\right)$$

and we can easily see that:

$$\mathbb{E}[X_i] = \frac{\alpha_i}{\sum_{j=1}^{n}\alpha_j} \tag{2.17}$$

As for any Beta(a, b) distribution, the mean is given by $\frac{a}{a+b}$.

Another property of the Dirichlet distribution that is used in derivations in this thesis, is the product moment generating function. It is expressed as follows.

Proposition 2.1 (Product moment generating function)
Let $\mathbf{X} \sim \text{Dirichlet}_n(\boldsymbol{\alpha})$ *on* $\mathbb{T}_n(1)$*. Let* $\mathbf{m}$ *be a n-dimensional vector with non-negative values. The product moment of* $\mathbf{X}$ *is given by:*

$$\mathbb{E}\left[\prod_{i=1}^{n}(\mathbf{X})_i^{m_i}\right] = \frac{\Gamma\left(\sum_{i=1}^{n}\alpha_i\right)}{\Gamma\left(\sum_{i=1}^{n}(m_i+\alpha_i)\right)}\prod_{i=1}^{n}\frac{\Gamma(m_i+\alpha_i)}{\Gamma(\alpha_i)} \tag{2.18}$$

To show that equation 2.18 is true, only a smart substitution is needed. Let us start with the definition of the expectation.

$$\begin{aligned}
\mathbb{E}\left[\prod_{i=1}^{n}(\mathbf{X})_i^{m_i}\right] &= \int\left(\prod_{i=1}^{n}x_i^{m_i}\right)\cdot\frac{\Gamma(\sum_{i=1}^{n}\alpha_i)}{\prod_{i=1}^{n}\Gamma(\alpha_i)}\prod_{i=1}^{n-1}x_i^{\alpha_i-1}\cdot(1-x_1-\cdots-x_{n-1})^{\alpha_n-1}\,dx_1\cdots dx_{n-1}\\
&= \frac{\Gamma(\sum_{i=1}^{n}\alpha_i)}{\prod_{i=1}^{n}\Gamma(\alpha_i)}\int\prod_{l=1}^{n-1}x_i^{m_i+\alpha_i-1}(1-x_1-\cdots-x_{n-1})^{m_n+\alpha_n-1}\,dx_1\cdots dx_{n-1}
\end{aligned}\tag{2.19}$$

To compute this integral, we need to do the same trick as in 2.16, but now iteratively. Substitute:
$x_{n-1} = (1 - x_1 - \cdots - x_{n-2})u_{n-1}$, $x_{n-2} = (1 - x_1 - \cdots - x_{n-3})u_{n-2}$, and so on till $x_2 = (1 - x_1)u_2$.

$$
\begin{aligned}
\mathbb{E}\left[\prod_{i=1}^{n}(\mathbf{X})_i^{m_i}\right] &= \frac{\Gamma(\sum_{i=1}^{n}\alpha_i)}{\prod_{i=1}^{n}\Gamma(\alpha_i)}\int\prod_{l=1}^{n-1}x_i^{m_i+\alpha_i-1}(1-x_1-\cdots-x_{n-1})^{m_n+\alpha_n-1}\,dx_1\cdots dx_{n-1}\\
&= \frac{\Gamma(\sum_{i=1}^{n}\alpha_i)}{\prod_{i=1}^{n}\Gamma(\alpha_i)}\int_0^1 x_1^{m_1+\alpha_1-1}(1-x_1)^{\sum_{i=2}^{n}(m_i+\alpha_i)-1}\,dx_1\\
&\quad\cdot\int_0^1 u_2^{m_2+\alpha_2-1}(1-u_2)^{\sum_{i=3}^{n}(m_i+\alpha_i)-1}\,du_2\cdot\ldots\cdot\int_0^1 u_{n-1}^{m_{n-1}+\alpha_{n-1}-1}(1-u_{n-1})^{m_n+\alpha_n-1}\,du_{n-1}\\
&= \frac{\Gamma(\sum_{i=1}^{n}\alpha_i)}{\prod_{i=1}^{n}\Gamma(\alpha_i)}\cdot\mathrm{B}\left(m_1+\alpha_1,\sum_{i=2}^{n}(m_i+\alpha_i)\right)\cdot\mathrm{B}\left(m_2+\alpha_2,\sum_{i=3}^{n}(m_i+\alpha_i)\right)\cdot\ldots\cdot\mathrm{B}\left(m_{n-1}+\alpha_{n-1},m_n+\alpha_n\right)\\
&= \frac{\Gamma\left(\sum_{i=1}^{n}\alpha_i\right)}{\Gamma\left(\sum_{i=1}^{n}(m_i+\alpha_i)\right)}\prod_{i=1}^{n}\frac{\Gamma(m_i+\alpha_i)}{\Gamma(\alpha_i)}
\end{aligned}
\tag{2.20}
$$

Where $B(\cdot,\cdot)$ is the beta function, and in the last step, its gamma function representation from equation 2.4 is used.

Visualization

Dirichlet distributed random vectors live on a simplex $\mathbb{T}_n(1)$, such that a draw from the Dirichlet distribution results in a vector with probabilities that sum to 1. To get an idea of the Dirichlet density, it is necessary to understand the plane on which $\mathbf{X}$ can lie. For a three-dimensional $\mathbf{X}$, this is visualized in 2.2. In higher dimensions, a hyperplane will describe the simplex, but it cannot be intuitively visualized anymore.

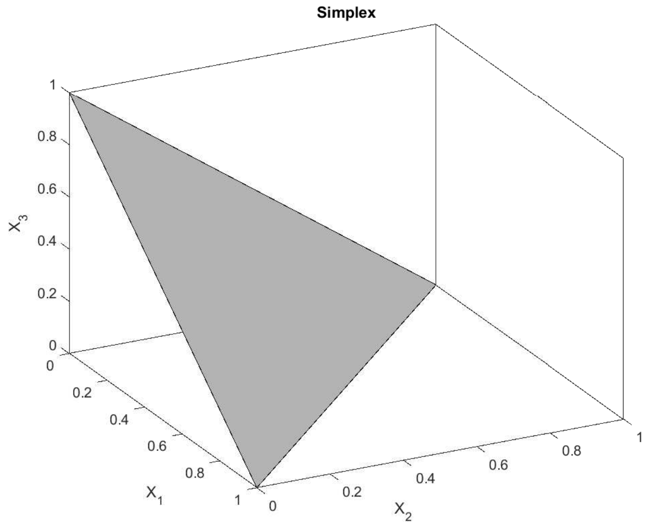
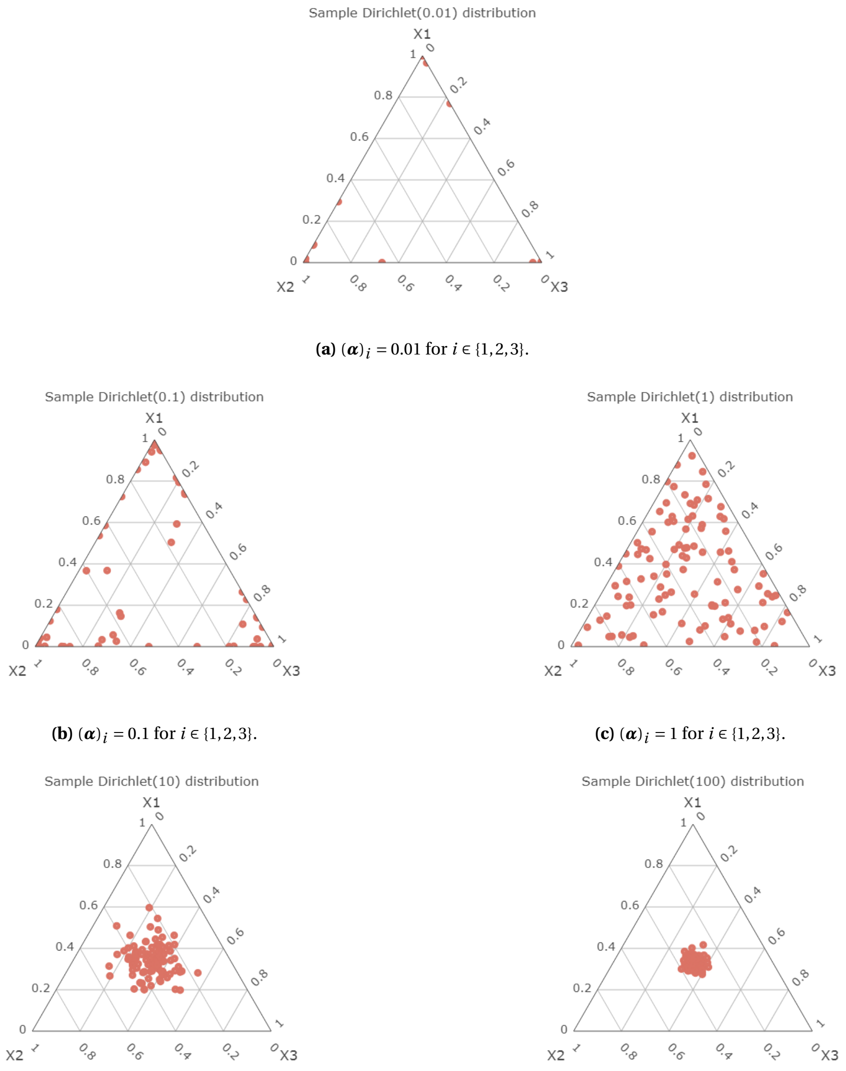


Figure 2.2: Three-dimensional simplex $\mathbb{T}_3(1)$. On the gray plane lie the values of X_1, X_2 and X_3 for $\mathbf{X}\in\mathbb{T}_3(1)$, such that the sum of X_1, X_2 and X_3 equal to 1.

To understand the Dirichlet density, many samples are drawn and visualized in a triangle format. The triangle from figure 2.2 is mapped to a two-dimensional representation in the plots in figure 2.3.
Different values for parameter vector $\boldsymbol{\alpha}$ of the Dirichlet distribution are chosen, however the symmetry remains, that is, all elements $(\boldsymbol{\alpha})_i$ are the same. The plots of samples of Dirichlet$(\boldsymbol{\alpha})$-distributions on a three-dimensional simplex for different values of $(\boldsymbol{\alpha})_i$ are shown in figure 2.3. Note that each dot represents a sample.



(a) $(\boldsymbol{\alpha})_i = 0.01$ for $i \in \{1,2,3\}$.

(b) $(\boldsymbol{\alpha})_i = 0.1$ for $i \in \{1,2,3\}$.

(c) $(\boldsymbol{\alpha})_i = 1$ for $i \in \{1,2,3\}$.

(d) $(\boldsymbol{\alpha})_i = 10$ for $i \in \{1,2,3\}$.

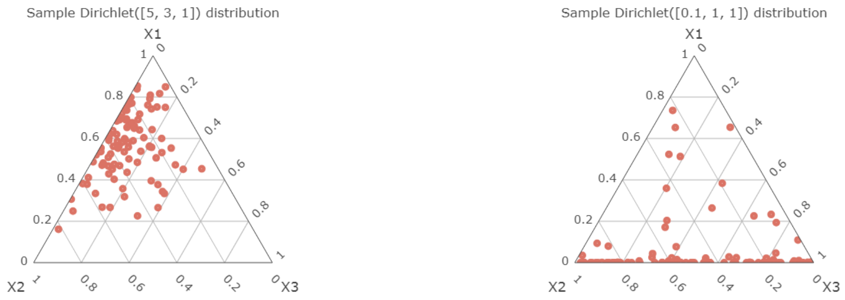
(e) $(\boldsymbol{\alpha})_i = 100$ for $i \in \{1,2,3\}$.

Figure 2.3: 100 samples from Dir(α) distribution with the same $(\boldsymbol{\alpha})_i$ for $i \in \{1,2,3\}$.

From figure 2.3 we can deduce a pattern. When parameters $(\boldsymbol{\alpha})_i,\ i \in \{1,2,3\}$ are all equal to 1, the distribution is in fact uniform, as can be seen in 2.3c. When $(\boldsymbol{\alpha})_i,\ i \in \{1,2,3\}$ are smaller than 1, there is a tendency versus one of the three dimensions. For every sample from the distribution, this can be a different dimension. In each sample the value of one of the X_i's is near 1, while the others are 0. This effect is the strongest for $(\boldsymbol{\alpha})_i = 0.01,\ i \in \{1,2,3\}$, as shown in 2.3a.
On the other hand, when the $(\boldsymbol{\alpha})_i, i \in \{1,2,3\}$ are larger than 1, the dots in the graph move towards the middle. This results in all samples lying in the middle of the triangle. Note that, by definition, the sum of the values of X_1, X_2 and X_3 is always equal to 1. The larger the $(\boldsymbol{\alpha})_i$, the more similar will be the values of X_1, X_2 and X_3.

Of course, the $(\boldsymbol{\alpha})_i, i \in \{1,2,3\}$ need not all have the same value. One can also take asymmetric Dirichlet priors. Samples are drawn from two asymmetric three-dimensional Dirichlet distributions and shown in figure 2.4.

The plots in figure 2.4 show very clearly the influence of the parameters $(\boldsymbol{\alpha})_i$ on the samples. In 2.4a, all samples have a strong tendency towards X_1, a less strong movement towards X_2 and a small tendency towards X_3. In this sense, the parameter vector $\boldsymbol{\alpha}$ can capture initial belief via the Dirichlet prior, because a larger $(\boldsymbol{\alpha})_i$ will in general result in a larger X_i. Note that the patterns of figure 2.3 are still valid. If you would take $\alpha_1 = 50, \alpha_2 = 30, \alpha_3 = 10$, all dots will be closer to each other, but the spot will be centered mostly towards X_1, a little less towards X_2 and not more towards X_3 than shown in 2.3d.



(a) $\alpha_1 = 5, \alpha_2 = 3, \alpha_3 = 1$.

(b) $\alpha_1 = 0.1, \alpha_2 = 1, \alpha_3 = 1$.

Figure 2.4: 100 samples from asymmetric Dirichlet distributions.
The axis of X_1 is on the left, the one of X_3 on the right and of X_2 on below the triangle.

In 2.4b, the dots lie on the line between X_2 and X_3, as their corresponding parameters are highest, whereas α_1 is smaller and therefore results in smaller values of X_1.

2.3. Natural language processing

Natural language processing is an overlapping field in computer science, artificial intelligence, and linguistics, in which all kinds of processing of human languages are involved. Examples are predictive text generation, automatic text generation, handwriting recognition, machine translation, and text summarization [4, 15]. The latter is of interest to use, as LDA aims to retrieve information of a large data set of reviews and summarize people's opinions.

The field of natural language processing (NLP) is vast, and applications are numerous. In this thesis, NLP is used in the preprocessing steps in which the reviews are prepared to be analyzed by LDA. That is, the data needs to be cleaned before it can be used. To this end, we used a combination of KNIME [23], Microsoft Excel and Python software, to prepare all data using the following steps.

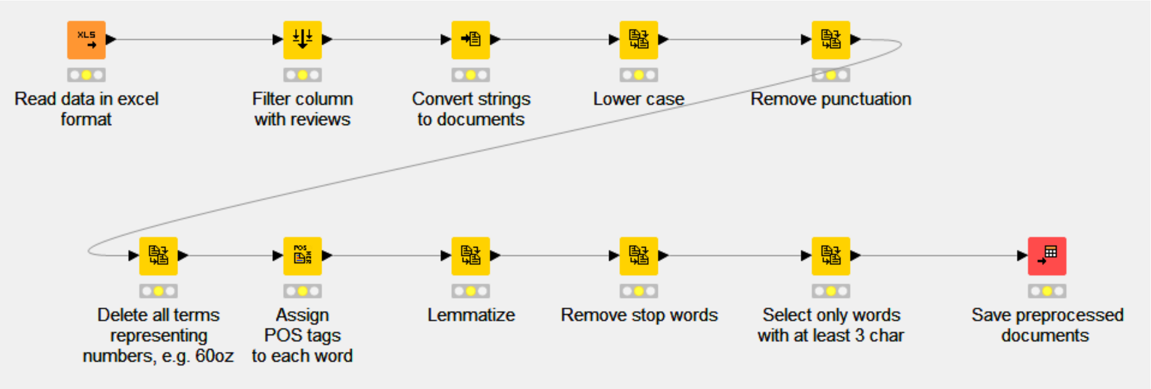

Read data in excel format
Filter column with reviews
Convert strings to documents
Lower case
Remove punctuation
Delete all terms representing numbers, e.g. 60oz
Assign POS tags to each word
Lemmatize
Remove stop words
Select only words with at least 3 char
Save preprocessed documents

Figure 2.5: Data preprocessing workflow in KNIME.

First, only review text needs to be retrieved, so all other information that might be in the data set, such as e-mail addresses or websites need to be removed. Then, it is made sure that the data is in the right data format. In KNIME, this is the 'document' class, while in Python we use lists with strings, in which each string contains a review. Subsequently, all capital letters are converted to lower case letters. In this way, words as *Like* and *like* are considered the same in the model, as they should be. The next step is to replace all punctuation and special symbols. These are not needed in the analysis. Note that also apostrophes are removed, such that the words *doesn't* become *doesnt*. Because this preprocessing step is applied consistently, we know that all *doesnt* words used to be *doesn't*. Another remark needs to be made, as in the extension of LDA introduced in this

thesis (see chapter 6), sentences and phrases are needed. Therefore, when the extended version of LDA is applied to the data set, commas, periods, question marks, exclamation marks, parentheses, (semi)colons and brackets are left in the data because these are used later on to split reviews into phrases.
In the data preparation process of both versions of LDA, numbers, and words containing numbers are removed. Consequently, the toughest NLP step is applied to the data: POS-tagging and lemmatization. POS stands for part-of-speech and is the function of a word in a sentence. Different programs are developed to automatically assign a part-of-speech to each word in the document. With the POS-tag of each word and the words themselves, lemmatization can be done. This is a process of truncating each word to a root. Consider the word *walking*. The POS-tag of this word indicates that it is a verb and that it is in the present continuous form. Therefore, its lemma is *walk*. By lemmatization of all words, we reduce the size of the vocabulary (the total number of unique words in the data set), and analyses are improved, as verbs that are conjugated differently are considered the same after the lemmatization steps. Moreover, adverbs and adjectives that come from the same lemma are considered equal.
Another preprocessing step that helps to reduce the vocabulary size is the removal of stop words. There are many lists with English stop words available in software or online containing words as *the*, *a*, *it*, *though*, et cetera. The stop word list used in this thesis can be found in the section B.5 in the appendix. These are uninformative in the analysis and only unnecessarily increase the size of the data set. Therefore, removal of stop words is often applied. In addition, short words can be removed. Because it is useful in opinion mining to leave the word ‘not’, it is chosen to only remove one and two letter words. The last step, before inserting the data in the LDA model, is the removal of low-frequency words. The size of the vocabulary determines the size of the parameters that need to be estimated in the model. For this reason, it is essential that only words that contribute to the analysis are contained in the vocabulary. Words that occur very rarely will not have a great contribution to the results of LDA, so they are discovered using frequency counts and then removed from the entire data set.

With the wholly cleaned and reduced data set consisting of lists of strings with reviews, Latent Dirichlet Allocation can be done, as is explained in chapter 3.

2.4. Model selection criteria

In every statistical model in which inference is done and parameters are estimated, model validation is needed. We need to check if the estimated parameters are good, but what is good? In this section, two methods to value the parameter estimations are explained.

When looking at the quality of the inferred model parameters in topic modelling, information theory comes into place. Many measures to quantify the goodness of fit originate from information theory. The most fundamental element in this field of science is the Shannon information, introduced by Claude Shannon in 1948[2].

Definition 2.3 (Shannon information [30])
The Shannon information content of an outcome x is defined to be:

$$h(x) = \log_2 \frac{1}{\mathbb{P}(x)}$$

With $\mathbb{P}(x)$ the probability of x and $h(x)$ measured in bits.

When looking at all possible outcomes that a random variable can have, the entropy or weighted average of the Shannon information comes into place. It is defined in [30] for an ensemble, which is just a random variable X with outcome space Ω_X and corresponding probabilities collected in $\mathscr{P}_X$.

Definition 2.4 (Entropy)
The entropy of an ensemble $X = (x, \Omega_X, \mathscr{P}_X)$ with probability measure $\mathbb{P}$ is defined to be the average Shannon information content of an outcome:

$$H(X) = \sum_{x \in \Omega_X} \mathbb{P}(x) \log \frac{1}{\mathbb{P}(x)} \tag{2.21}$$

[2] Did you know that Claude Shannon and Alan Turing, the inventor of the computer, had lunch together?

Here, capital X is used to denote the fact that entropy is computed of a discrete random variable X, with sample space Ω_X and probability measure $\mathbb{P}$. If $\mathbb{P}(x) = 0$ for some $x \in \Omega_X$, then $\mathbb{P}(x)\log\frac{1}{\mathbb{P}(x)}$ is defined to be equal to 0. Furthermore $H(X)$ is measured in bits, and is also referred to as the uncertainty of X.

The idea of entropy can best be understood when considering the example of flipping a coin again. First, assume that we have a fair coin, such that the probability of heads and tails is equal: $\mathbb{P}(H) = (T) = \frac{1}{2}$. Substituting this in equation 2.21 with X being the random variable with sample space $\{H, T\}$ and the aforementioned probabilities, we get $H(X) = \log_2(2) = 1$. This means that we need only 1 bit to communicate the outcome of the coin flip, namely 1 for heads and 0 for tails (or vice versa). In the same way for a 4-sided dice with 4 different outcomes, we need 2 bits, as $H(X) = \frac{1}{4}\log_2(4) + \frac{1}{4}\log(4) + \frac{1}{4}\log(4) + \frac{1}{4}\log(4) = 2$. However, if we have a strange dice, with 1 unique side (e.g. 1) and 3 sides that show the same number (e.g. 2), the probabilities and the entropy will change: $H(X) = \frac{1}{4}\log_2(4) + \frac{3}{4}\log_2(\frac{4}{3}) \approx 0.81$. Note that the entropy is lower than for the fair dice, where we needed 2 bits to communicate the result. One can think of this result as if more information is already hidden in the outcome and does not have to be communicated, so only '0.81' bit is needed to tell the result of the throw to your opponent. This result is in general true, as is stated in the second item of the highlighted properties of the entropy from [30].

Theorem 2.2 (Properties entropy)

- *$H(X) \geq 0$, with equality if and only if $\exists i$ such that $p_i = \mathbb{P}(X = a_i) = 1$*
- *$H(X)$ is maximized if $\mathbf{p} = (p_1, \dots p_I)$ is uniform. That is if $p_i = \frac{1}{|\Omega_X|}, \forall i \in \{1, \dots I\}$. Then $H(X) = \log(|\Omega_X|)$. In general, we have $H(X) \leq \log(|\Omega_X|)$.*
- *The joint entropy of random variables X and Y with sample spaces Ω_X and Ω_Y and joint probability measure $\mathbb{P}$, is defined as:*

$$H(X, Y) = \sum_{x\in\Omega_X, y\in\Omega_Y} \mathbb{P}(x, y)\log\frac{1}{\mathbb{P}(x, y)}$$

  and if X and Y independent random variables, then $H(X, Y) = H(X) + H(Y)$.

A metric that is often used for the comparison of two probability distributions is the Kullback-Leibler divergence. In the field of information theory, it is called the relative entropy. Note that it is not an actual distance in the mathematical sense.

Definition 2.5 (Relative entropy, KL-divergence)
The relative entropy, also called the Kullback-Leibler divergence, between two discrete probability distributions $\mathbf{p}$ and $\mathbf{q}$ that are defined over the same sample space Ω_X is given by:

$$D_{KL}(\mathbf{p}\|\mathbf{q}) = \sum_{x\in\Omega_X} p(x)\log\frac{p(x)}{q(x)} \tag{2.22}$$

To give an idea of the working principle of this relative entropy, we consider a small example. Let $\mathbf{p} = (0.6, 0.2, 0.1, 0.05, 0.05)$ and $\mathbf{q} = (0.6, 0.2, 0.05, 0.05, 0.1)$. The relative entropy is then $D_{KL}(\mathbf{p}\|\mathbf{q}) = 0.05 \cdot \log(2) \approx 0.03$. The only differences between $\mathbf{p}$ and $\mathbf{q}$ are the swapped probabilities of the third and fifth element, which have both already small probability mass. If we would halve the first element and triple the third element of $\mathbf{p}$ to get $\mathbf{q}$, i.e. $\mathbf{q} = (0.3, 0.2, 0.3, 0.05, 0.05)$, the relative entropy will be $D_{KL}(\mathbf{p}\|\mathbf{q}) = 0.6 \cdot \log(2) - 0.1 \cdot \log(3) \approx 0.31$, which is a lot higher than the previous score. As expected, large changes in probability mass (in the absolute sense) result in larger KL-divergence scores than small changes (in the absolute sense).

The relative entropy is also defined for the comparison of densities of two continuous random variables sharing the same domain. Then, the sum in the definition above becomes an integral over the domain, and the probability densities replace the probability mass functions. For the qualification of our model parameters, we cannot compare the estimated distributions $\mathbf{q}$ with the true distribution $\mathbf{p}$, as the true distribution is unknown. Nevertheless, the Kullback-Leibler divergence is used for other purposes in chapter 7 and section 4.2.1.

In topic modelling, another statistic is used for model comparison: the perplexity. In the field of NLP, and language and topic models (such as LDA), this measure is most frequently used to observe the difference in the quality of the model when parameters like the number of topics, the vocabulary size or the number of iterations are changed. The model with the lowest perplexity is then assumed to be the best fit on the data.

The definition of the perplexity is taken from the original LDA paper [7] by Blei et al..

Definition 2.6 (Perplexity)
Consider a model that is trained on training data $\mathbf{w}_{\text{train}}$ *to obtain estimates for the model parameters. Then, the perplexity of the left-out test data set* $\mathbf{w}_{\textbf{test}}$ *is defined as:*

$$\begin{aligned}\text{Perplexity}(\mathbf{w}_{\textbf{test}}) &= 2^{-\frac{\log_2(\mathbb{P}(\mathbf{w}_{\textbf{test}}))}{|\mathbf{w}_{\textbf{test}}|}} \\ &= e^{-\frac{\log(\mathbb{P}(\mathbf{w}_{\textbf{test}}))}{|\mathbf{w}_{\textbf{test}}|}}\end{aligned} \tag{2.23}$$

Where $|\mathbf{w}_{\textbf{test}}|$ *is the size of the test set.*

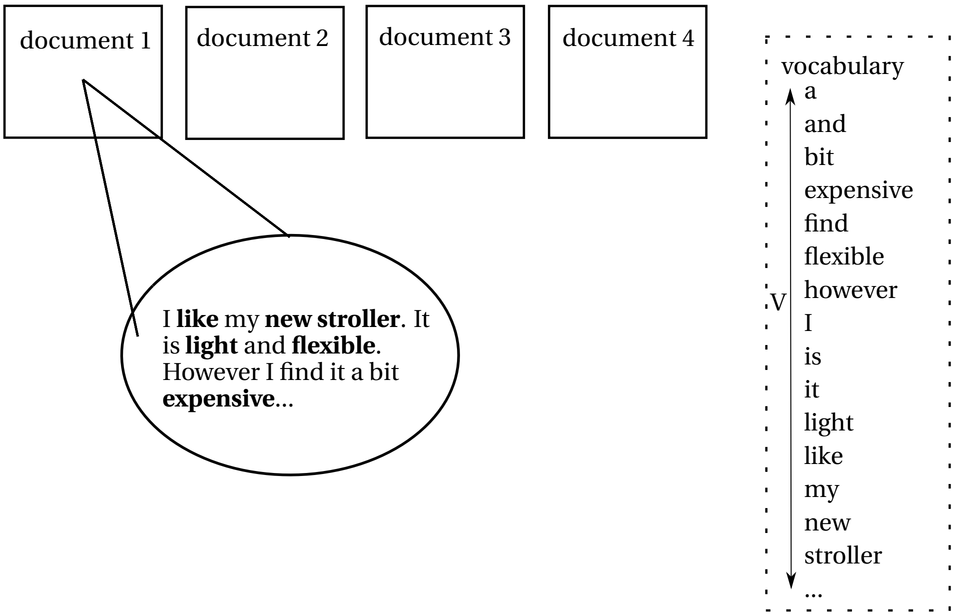
One can think of the perplexity as a comparison of the inferred model with the case of a uniform distribution. Remember that in the latter case, the entropy was highest, so in the perplexity, we observe to what extent the model has improved on the uninformative prior. Because the perplexity is only used for comparison among models or parameter settings, the 'best' model is the one that has the lowest entropy and thus retrieved the most information from the data.

3

Latent Dirichlet Allocation

In this thesis, we focus on the model called 'Latent Dirichlet Allocation'. This model was introduced by Blei et al. in 2003, and is essentially a hierarchical model that brings structure in a (large) set of documents. First the terminology used in the model and throughout this literature study must be set straight. Let there be a set of documents $\mathcal{D} = \{1, \ldots, M\}$, also referred to as the corpus. In this research, documents are customer reviews, but all kinds of text can be used as input for LDA. Each document $d \in \mathcal{D}$ in the corpus contains a list of words, represented with vector $\mathbf{w}_d$. Each $\mathbf{w}_d$ has its own length N_d, meaning that the documents in the corpus are of varying lengths. Furthermore, there is a finite set of (unique) words that occur in the corpus, conveniently called the vocabulary. The size of the vocabulary is denoted with V.

A simple example of 4 documents is shown below. Document 1 consists of 17 words, therefore $N_1 = 17$. Documents 2, 3 and 4 can have different lengths. The vocabulary is shown for only the words of document 1. Assuming that different words are used in documents 2, 3 and 4, the vocabulary size is $V > 17$. The words that indicate the writer's opinion are shown in boldface. These are the possible words of interest, since they contain a customer's opinion.



Each word $(\mathbf{w_d})_i$ in a document has two indices d and i, meaning that it is a word from document d and on location i within the document (with $d \in \{1, \ldots, M\}$ and $i \in \{1, \ldots, N_d\}$). The vocabulary consists of all words that occur in the corpus in alphabetic order and assigns to each word an index. The first word in the vocabulary

above, *a*, has index 1, the second word, *and*, is represented by 2, et cetera. A word $(\mathbf{w_d})_i$ is thus not given by its textual representation, e.g. *flexible*, but by its index in the vocabulary, 6. As a result, we know that $\forall d, i, (\mathbf{w_d})_i \in \{1, \ldots, V\}$.
Furthermore, the bag-of-words representation of documents in used in LDA. In this representation, word order is disregarded, so only the frequency of word occurrence in each document matters.

At last, we assume that there are K topics hidden in the corpus. These topics can be seen as common themes that can be found in reviews. In the example above, we see that the writer of document 1 writes about the lightness and flexibility of his/her new stroller. Also, he/she finds it expensive. There might be more people who write about flexibility, so then it becomes a theme. Besides, if many customers write 'I like this stroller', a topic consisting of the main word 'like' will be formed. Note that topics are not found by a label or overarching theme like 'comfort'; only a topic-word distribution rolls out of the algorithm. That is, each topic $k \in \{1, \ldots, K\}$ has a corresponding topic-word distribution, with higher probabilities for words that are important to this topic. The topic that is manually labeled to be about flexibility will have a high probability for the word 'flexible' in its topic-word distribution.
Note that in LDA, it is unknown beforehand what topics can be found in the review data set, as it is an unsupervised method. Even the number of topics, K, is unknown and must be determined by domain knowledge, size of the data set, and trial and error (which model gives the best fit based on a goodness-of-fit statistic). That is a small data set of M documents can barely give accurate results if $K \approx M$.

An overview of all sets, variables, and parameters is given in table 3.1.

Table 3.1: Overview of parameters and observational data in Latent Dirichlet Allocation.

Variable name	Meaning
$\mathscr{D}$	Set of documents, 'Corpus'
M	Number of documents
$\mathbf{w}_d$	List of words in document d
$(\mathbf{w_d})_i$	Word on location i in document d
N_d	Number of words in document d
V	Size of vocabulary, i.e. number of unique words in corpus
K	Number of topics
k	Topic index

3.1. Into the mind of the writer: generative process

The goal of LDA is to extract topics from a set of reviews via a hierarchical Bayesian model. Before we look at statistical inference methods (see chapter 4), we need to understand the hierarchy of the Bayesian model. The generative process that forms this hierarchy aims to summarize the writing process in the minds of the writers. To stay with the example of strollers, imagine you have just bought a new, very expensive buggy. As it has cost you a lot of money, you have high expectations, but the stroller turns out to be a bit disappointing. You want to share your experience with other customers, so you decide to write a review. The process that follows is the generative process, as you are going to generate a document. Latent Dirichlet Allocation assumes that the generative process goes as follows.

First, you think about which aspects you want to write. You feel disappointed, as the stroller you have bought was very expensive. Furthermore, you want to explain your disappointment: the stroller is very heavy, too large to fit in the car, and the basket underneath is too small. Thus, you want to talk about four topics: value for money, weight, size, and the basket. Of course, the labels that are now manually assigned to the topics do not necessarily occur explicitly in the reviews. You find your disappointment in the performancce of stroller compared to the price you paid for it the most essential aspect, so 40% of the words in the review are about this topic. The other three topics are formed by the rest of the document, with an equal number of words. That is, 20% of the words are about the weight, 20% about the size, and 20% about the basket. After this decision, you need to find the right words to describe your opinion. For each topic, there is a set of words in your own English vocabulary from which can be chosen. For example, for topic 'size' can be thought of: large, small, big, little, fit, huge, size, proportions, width, broad, height, et cetera. These sets of words exist for each topic about which you want to write. All these aspect words are then glued together with verbs, personal pronouns, and determinants to form a review of clear and correct English.

Mathematically speaking, this generative process of writing a review can be summarized in a hierarchical Bayesian model, with a prior belief on how the writer chooses its topics and a prior belief on which words occur in the set of words to choose from for each possible topic. The following scheme from [7] summarizes it.

1. For each document $d \in \{1,\dots,M\}$,
   draw a topic distribution parameter vector $\mathbf{\Theta_d}$ from a Dirichlet($\boldsymbol{\alpha}$) distribution, i.e. $\mathbf{\Theta_d} \sim$ Dirichlet($\boldsymbol{\alpha}$)
2. For each topic $k \in \{1,\dots,K\}$,
   draw a topic-word distribution parameter vector $\mathbf{\Phi_k}$ from a Dirichlet($\boldsymbol{\beta}$) distribution, i.e. $\mathbf{\Phi_k} \sim$ Dirichlet($\boldsymbol{\beta}$)
3. For each word i in document d,
   (a) Draw a topic $(\mathbf{Z_d})_i$ from a Multinomial(1,$\mathbf{\Theta_d}$)
   (b) Draw a word $(\mathbf{W_d})_i$ from a Multinomial(1,$\mathbf{\Phi_{(z_d)_i}}$)

Attention must be paid to steps 3a and 3b, because drawing from a Multinomial$(1, \mathbf{\Theta_d})$ results in drawing a vector instead of an integer. Therefore we define $\mathbf{\tilde{Z}_{d,i}} \sim$ Multinomial$(1,\mathbf{\Theta_d})$, such that $(\mathbf{z_d})_i = k \iff \mathbf{\tilde{z}_{d,i}} = (0,0,\dots,1,0,\dots,0)$ with only one 1 on the k-th dimension of $\mathbf{\tilde{z}_{d,i}}$. That is $\mathbf{\tilde{z}_{d,i}}$ is the unit vector in dimension k. So when it is written in this thesis that $(\mathbf{Z_d})_i$ is drawn from Multinomial(1,$\mathbf{\Theta_d}$), actually $\mathbf{\tilde{Z}_{d,i}}$ is drawn from Multinomial$(1,\mathbf{\Theta_d})$ and the mapping $(\mathbf{\tilde{z}_{d,i}})_k = 1 \Rightarrow (\mathbf{z_d})_i = k$ for some $k \in \{1,\dots,K\}$ is applied.
A similar definition can be made for the words: $(\mathbf{w_d})_i = k \iff (\mathbf{\tilde{w}_{d,i}})_k = 1$ for $\mathbf{\tilde{W}_{d,i}} \sim$ Multinomial$(1,\mathbf{\Phi_{(z_d)_i}})$. Consequently, the same steps are applied when we 'draw' $(\mathbf{W_d})_i$ from Multinomial(1,$\mathbf{\Phi_{(z_d)_i}}$).

In the generative process, some critical assumptions on independence are made. Each document-topic distribution $\mathbf{\Theta_d}$ is drawn independently from all other $\mathbf{\Theta_i}$ for $i \neq d$. This is reasonable, as it is probable that the writers of the reviews decide about which topics they want to write independently. It is thus assumed that there has been no communication between them beforehand. However, all customers write about the same product, so independence is a strong assumption that is expected not to be satisfied in each data set.
The same is valid for the topic-word distributions: each $\mathbf{\Phi_k}$ is drawn independently from all other $\mathbf{\Phi_j}$ for $j \neq k$. That is, the word probabilities that belong to a particular topic are independent of the word probability distributions of other topics.
Furthermore, all $\mathbf{\Theta}$ and $\mathbf{\Phi}$ are independent by construction, which makes sense, as the topic distribution of a certain review and the sets of words to choose from per topic have nothing to do with each other.
Lastly, each topic $(\mathbf{Z_d})_i$ is drawn independently from the corresponding Multinomial(1,$\mathbf{\Theta_d}$) and therefore each pair $((\mathbf{Z_d})_i, (\mathbf{W_d})_i)$ is independent from every other pair $\big((\mathbf{Z_i})_j, (\mathbf{W_i})_j\big)$. These assumptions will be used later on in statistical inference on the hierarchical model.

The generative process can be visualized in a plate notation, shown in figure 3.1. One should read figure 3.1 as follows. The three rectangles can be read as three 'for loops' and they represent three levels in the hierarchical model. First consider the two on the right of figure 3.1. The outer rectangle represents the corpus or the loop over documents. The hyperparameter vector $\boldsymbol{\alpha}$ is outside the rectangle and is therefore independent of the documents. Random vector $\mathbf{\Theta}$ is within the loop, so this vector is drawn for each document. One level deeper, we look at the word in the document. For each word instance in the document, we first draw a topic and then a word. These draws are done as often as there are words in the document, therefore N_d times for document d.
The rectangle on the left is separate and does not depend directly on the documents. The hyperparameter vector $\boldsymbol{\beta}$ is outside the rectangle, meaning that this prior belief on the topic-word distributions $\mathbf{\Phi}$ is the same for each topic. Then, a random vector $\mathbf{\Phi}$ is drawn K times, as is denoted in the corner. For clarity, the indices are left out in figure 3.1.
Furthermore, the gray variable in the plate notation represents the word random variable W, which is actually observed. The circles with $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are dotted because they are pre-set values and thus constant. These are located in the top of our hierarchical scheme. No further distribution is imposed on either $\boldsymbol{\alpha}$ or $\boldsymbol{\beta}$ in the basic LDA model.

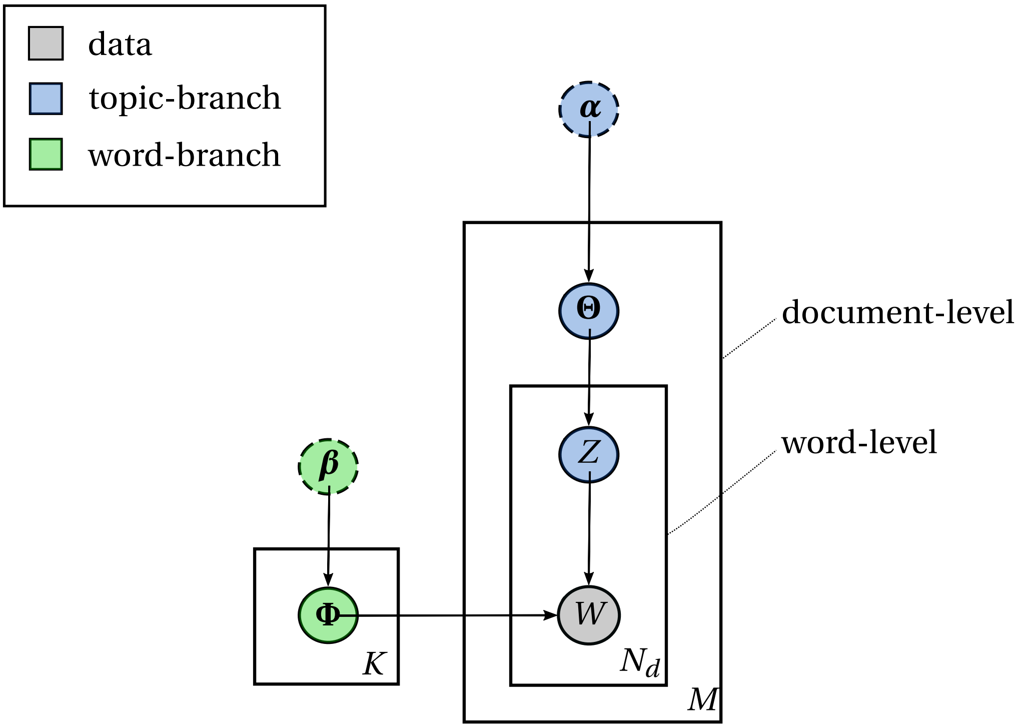


Figure 3.1: Plate notation of Latent Dirichlet Allocation as visualized in [25]. The hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are denoted with a dotted circle. $\boldsymbol{\Theta}$ represents the document-topic distribution, and $\boldsymbol{\Phi}$ is the topic-word distribution. Random variable Z is the topic that has one-to-one correspondence with word W.

An overview of all parameters and random variables, their dimensions, and the spaces in which they exist, is made in table 3.2.

Table 3.2: Random variables and constants used in Latent Dirichlet Allocation.

Symbol	Meaning	Type (and size)	Space
V	Size of vocabulary	integer	$\mathbb{N}$
K	Number of topics	integer	$\mathbb{N}$
M	Number of documents in corpus	integer	$\mathbb{N}$
$\boldsymbol{\alpha}$	Prior belief on document-topic distribution (see section 2.2)	vector: $1 \times K$	$\mathbb{R}^K_{>0}$
$\boldsymbol{\beta}$	Prior belief on topic-word distribution (see section 2.2)	vector: $1 \times V$	$\mathbb{R}^V_{>0}$
$\boldsymbol{\Phi}_k$	Parameter of multinomial word distribution for topic k	vector: $1 \times V$	$\mathbb{T}_V(1)$, (simplex)
$\boldsymbol{\Theta}_d$	Parameter vector of multinomial topic distribution for document d	vector: $1 \times K$	$\mathbb{T}_K(1)$, (simplex)
$\tilde{\mathbf{z}}_{d,i}$	Unit vector in the dimension of the chosen topic corresponding to word (d,i)	vector: $1 \times K$	$\{0,1\}^K$
$(\mathbf{z_d})_i$	Topic (index) for word i in document d	integer	$\{1,\ldots,K\}$
$\tilde{\mathbf{w}}_{d,i}$	Unit vector in the dimension of the chosen word (index)	vector: $1 \times V$	$\{0,1\}^V$
$(\mathbf{w_d})_i$	Word index i in document d	integer: 1×1	$\{1,\ldots,V\}$

3.2. Important distributions in LDA

In the generative process, several probability distributions are mentioned. In this section, the distribution of each random variable in the scheme in figure 3.1 conditional on its previous node is given. Note that $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the hyperparameters, and no distribution on these is imposed.

Starting in the top of the scheme with the topic distribution per document. We know that the parameter vector $\mathbf{\Theta_d}$ of document-topic distribution is Dirichlet distributed for each document $d \in \{1,\dots,M\}$, that is:

$$
\begin{aligned}
(\mathbf{\Theta_d}|\boldsymbol{\alpha}) &\sim \text{Dirichlet}(\boldsymbol{\alpha}) \\
p(\boldsymbol{\theta}_\mathbf{d}|\boldsymbol{\alpha}) &= \frac{\Gamma\left(\sum_{k=1}^{K}(\boldsymbol{\alpha})_k\right)}{\prod_{k=1}^{K}\Gamma((\boldsymbol{\alpha})_k)} \prod_{k=1}^{K} (\boldsymbol{\theta}_\mathbf{d})_k^{(\boldsymbol{\alpha})_k-1}
\end{aligned}
\tag{3.1}
$$

Throughout this thesis, the notation $(\mathbf{\Theta_d})_k$ is used for the k-th element of vector $\mathbf{\Theta_d}$ (in boldface).

Also the word distribution per topic, $\mathbf{\Phi_k}$, is Dirichlet distributed with parameter vector $\boldsymbol{\beta}$ for each $k \in \{1,\dots,K\}$, that is:

$$
\begin{aligned}
\left(\mathbf{\Phi_k}|\boldsymbol{\beta}\right) &\sim \text{Dirichlet}(\boldsymbol{\beta}) \\
p(\boldsymbol{\phi}_\mathbf{k}|\boldsymbol{\beta}) &= \frac{\Gamma\left(\sum_{j=1}^{V}(\boldsymbol{\beta})_j\right)}{\prod_{j=1}^{V}\Gamma((\boldsymbol{\beta})_j)} \prod_{j=1}^{V} (\boldsymbol{\phi}_\mathbf{k})_j^{(\boldsymbol{\beta})_j-1}
\end{aligned}
\tag{3.2}
$$

Topic $\mathbf{\tilde{Z}_{d,i}}$ is drawn from a Multinomial distribution with parameter $\mathbf{\Theta_d}$, also from document d:

$$
\begin{aligned}
\left(\mathbf{\tilde{z}_{d,i}}|\mathbf{\Theta_d}\right) &\sim \text{Multinomial}(1,\mathbf{\Theta_d}) \\
p(\mathbf{\tilde{z}_{d,i}}|\boldsymbol{\theta}_\mathbf{d}) &= \frac{\Gamma\left(\sum_{k=1}^{K}(\mathbf{\tilde{z}_{d,i}})_k+1\right)}{\prod_{k=1}^{K}\Gamma((\mathbf{\tilde{z}_{d,i}})_k+1)} \prod_{k=1}^{K} (\boldsymbol{\theta}_\mathbf{d})_k^{(\mathbf{\tilde{z}_{d,i}})_k} \\
&= \prod_{k=1}^{K} (\boldsymbol{\theta}_\mathbf{d})_k^{(\mathbf{\tilde{z}_{d,i}})_k} = (\boldsymbol{\theta}_\mathbf{d})_{(\mathbf{z_d})_i}
\end{aligned}
\tag{3.3}
$$

Note that the probability of $(\mathbf{Z_d})_i$ being topic l is equal to the l-th element of document-topic vector $\mathbf{\Theta_d}$. This is a very natural way to consider the topic probabilities.

A similar procedure can be followed for the word probability density given that the corresponding topic $(\mathbf{Z_d})_i = k$ for $k \in \{1,\dots,K\}$:

$$
\begin{aligned}
\left(\mathbf{\tilde{W}_{d,i}}|(\mathbf{Z_d})_i = k,\mathbf{\Phi_k}\right) &\sim \text{Multinomial}(1,\mathbf{\Phi_k}) \\
p(\mathbf{\tilde{w}_{d,i}}|(\mathbf{z_d})_i = k,\boldsymbol{\phi}_\mathbf{k}) &= \frac{\Gamma\left(\sum_{j=1}^{V}(\mathbf{\tilde{w}_{d,i}})_j+1\right)}{\prod_{j=1}^{V}\Gamma((\mathbf{\tilde{w}_{d,i}})_j+1)} \prod_{j=1}^{V} (\boldsymbol{\phi}_\mathbf{k})_j^{(\mathbf{\tilde{w}_{d,i}})_j} \\
&= \prod_{j=1}^{V} (\boldsymbol{\phi}_\mathbf{k})_j^{(\mathbf{\tilde{w}_{d,i}})_j} = \left(\boldsymbol{\phi}_\mathbf{k}\right)_{(\mathbf{w_d})_i}
\end{aligned}
\tag{3.4}
$$

Again, given that you chose topic $(\mathbf{Z_d})_i = k$, the probability of picking the j-th word in the vocabulary, i.e. $\mathbb{P}((\mathbf{w_d})_i = j|(\mathbf{z_d})_i = k,\boldsymbol{\phi}_\mathbf{k})$, is just equal to the j-th element of topic-word probability vector $\mathbf{\Phi_k}$.

3.3. Probability distribution of the words

The probability of having the corpus as observed given the hyperparameters, is given by $p(\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta})$. To obtain a closed-form expression for this 'likelihood', we first derive $p(\mathbf{w},\mathbf{z}|\boldsymbol{\alpha},\boldsymbol{\beta})$. In the derivation the conditioning on the hyperparameters is omitted in the notation, as this is trivial.

Let us take a document d, so we consider the case in which we only have **one document**, d. The document-topic distribution for this document is denoted $\mathbf{\Theta_d}$ and the topic-word distributions are denoted by $\mathbf{\Phi_k}$ for

$k = 1,\ldots,K$, as shown in table 3.2. We want to know the joint distribution of all words and corresponding topics in this document i.e. $(\mathbf{w},\mathbf{z}) = (w_1,\ldots w_{N_d}, z_1,\ldots,z_{N_d})$.

$$\begin{aligned}
p(\mathbf{W}=\mathbf{w},\mathbf{Z}=\mathbf{z}) &= \mathbb{E}\left[\mathbb{1}_{\{\mathbf{W}=\mathbf{w},\mathbf{Z}=\mathbf{z}\}}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\mathbb{1}_{\{\mathbf{W}=\mathbf{w},\mathbf{Z}=\mathbf{z}\}}|\boldsymbol{\Theta}_\mathbf{d},\boldsymbol{\Phi}\right]\right] \\
&= \mathbb{E}\left[p\left(\mathbf{W}=\mathbf{w},\mathbf{Z}=\mathbf{z}|\boldsymbol{\Theta}_\mathbf{d},\boldsymbol{\Phi}\right)\right] \\
&= \mathbb{E}\left[\prod_{i=1}^{N_d} p\left(W_i = w_i, Z_i = z_i|\boldsymbol{\Theta}_\mathbf{d},\boldsymbol{\Phi}\right)\right] \qquad * \\
&= \mathbb{E}\left[\prod_{i=1}^{N_d} p\left(W_i = w_i|Z_i = z_i,\boldsymbol{\Theta}_\mathbf{d},\boldsymbol{\Phi}\right)\cdot p\left(Z_i = z_i|\boldsymbol{\Theta}_\mathbf{d},\boldsymbol{\Phi}\right)\right] \\
&= \mathbb{E}\left[\prod_{i=1}^{N_d} (\boldsymbol{\Phi}_{\mathbf{z_i}})_{w_i}\cdot(\boldsymbol{\Theta}_\mathbf{d})_{z_i}\right]
\end{aligned} \tag{3.5}$$

$*$ this can be done because each topic and word combination, i.e., (W_i, Z_i), is drawn from respectively the Multinomial($\boldsymbol{\Theta}_\mathbf{d}$) and Multinomial($\boldsymbol{\Phi}_{z_i}$) distributions, independently from all other pairs.

Note that for each i in the last line of expression 3.5, $(\boldsymbol{\Phi}_{\mathbf{z_i}})_{w_i}$ and $(\boldsymbol{\Theta}_\mathbf{d})_{z_i}$ are independent random variables by construction, as shown in the plate notation of the hierarchical model in figure 3.1 and in the generative process. All $(\boldsymbol{\Phi}_\mathbf{i})_j$ are drawn from their prior distribution, independently from $\boldsymbol{\Theta}_\mathbf{d}$ for all $d \in \{1,\ldots,M\}$. Therefore the expectation in 3.5 can be split up. Remember that among the $(\boldsymbol{\Phi}_{\mathbf{z_i}})_{w_i}forj \in \{1,\ldots,N_d\}$ there is a dependence structure, in particular because $\sum_{j=1}^{V}(\boldsymbol{\Phi}_{\mathbf{z_i}})_j = 1$, so these cannot be split up. The same can be said about the $(\boldsymbol{\Theta}_\mathbf{d})_{z_i}$, because also $\sum_{k=1}^{K}(\boldsymbol{\Theta}_\mathbf{d})_k = 1$.

Continuing derivation 3.5:

$$\begin{aligned}
p(\mathbf{W}=\mathbf{w},\mathbf{Z}=\mathbf{z}) &= \mathbb{E}\left[\prod_{i=1}^{N_d} (\boldsymbol{\Phi}_{\mathbf{z_i}})_{w_i}\cdot(\boldsymbol{\Theta}_\mathbf{d})_{z_i}\right] \\
&= \mathbb{E}\left[\prod_{i=1}^{N_d} (\boldsymbol{\Phi}_{\mathbf{z_i}})_{w_i}\right]\cdot\mathbb{E}\left[\prod_{i=1}^{N_d} (\boldsymbol{\Theta}_\mathbf{d})_{z_i}\right] \\
&= \mathbb{E}\left[\prod_{k=1}^{K}\prod_{j=1}^{V} (\boldsymbol{\Phi}_\mathbf{k})_j^{(\mathbf{n_k})_j}\right]\cdot\mathbb{E}\left[\prod_{k=1}^{K} (\boldsymbol{\Theta}_\mathbf{d})_k^{(\mathbf{m})_k}\right] \\
&= \left(\prod_{k=1}^{K}\mathbb{E}\left[\prod_{j=1}^{V} (\boldsymbol{\Phi}_\mathbf{k})_j^{(\mathbf{n_k})_j}\right]\right)\cdot\mathbb{E}\left[\prod_{k=1}^{K} (\boldsymbol{\Theta}_\mathbf{d})_k^{(\mathbf{m})_k}\right] \qquad *
\end{aligned} \tag{3.6}$$

where we define $(\mathbf{m})_k$ as the number of times a word in document d is assigned to topic k and $(\mathbf{n_k})_j$ as the number of times a word in the document is assigned to topic k and the word is equal to word j in the vocabulary. Note that this is a logical thing to do, as the probability of a word occurring 5 times in a document is the probability of that word to the power 5 i.e. $(\mathbb{P}(\text{word}))^5$. At $*$ we can take the product over all topics out of the expectation, because for each topic, the vector $\boldsymbol{\Phi}_k$ is drawn independently from all other topics from the Dirichlet($\boldsymbol{\beta}$) distribution.

The only difficult expressions left are the product moments of respectively $\boldsymbol{\Phi}_k$ and $\boldsymbol{\Theta}_\mathbf{d}$: $\mathbb{E}\left[\prod_{j=1}^{V}(\boldsymbol{\Phi}_\mathbf{k})_j^{(\mathbf{n_k})_j}\right]$ and $\mathbb{E}\left[\prod_{k=1}^{K}(\boldsymbol{\Theta}_\mathbf{d})_k^{(\mathbf{m})_k}\right]$. We use the expression for the product moment of a Dirichlet distribution, as derived in section 2.2 in equation 2.18, to obtain the final result of the joint distribution of the word and topic vectors in document d.

$$
\begin{aligned}
p(\mathbf{W}=\mathbf{w},\mathbf{Z}=\mathbf{z}) &= \left(\prod_{k=1}^{K}\mathbb{E}\left[\prod_{j=1}^{V}(\boldsymbol{\Phi}_{\mathbf{k}})_j^{(\mathbf{n_k})_j}\right]\right)\cdot\mathbb{E}\left[\prod_{k=1}^{K}(\boldsymbol{\Theta}_{\mathbf{d}})_k^{(\mathbf{m})_k}\right]\\
&= \left(\prod_{k=1}^{K}\left[\frac{\Gamma\left(\sum_{i=1}^{V}(\boldsymbol{\beta})_i\right)}{\Gamma\left(\sum_{j=1}^{V}(\mathbf{n_k})_j+(\boldsymbol{\beta})_j\right)}\prod_{j=1}^{V}\frac{\Gamma((\mathbf{n_k})_j+(\boldsymbol{\beta})_j)}{\Gamma((\boldsymbol{\beta})_j)}\right]\right)\cdot\left[\frac{\Gamma\left(\sum_{i=1}^{K}(\boldsymbol{\alpha})_i\right)}{\Gamma\left(\sum_{i=1}^{K}(\mathbf{m})_i+(\boldsymbol{\alpha})_i\right)}\prod_{k=1}^{K}\frac{\Gamma((\mathbf{m})_k+(\boldsymbol{\alpha})_k)}{\Gamma((\boldsymbol{\alpha})_k)}\right]\\
&= \left(\frac{\Gamma\left(\sum_{i=1}^{V}(\boldsymbol{\beta})_i\right)}{\prod_{j=1}^{V}\Gamma((\boldsymbol{\beta})_j)}\right)^{K}\left[\prod_{k=1}^{K}\frac{\prod_{j=1}^{V}\Gamma((\mathbf{n_k})_j+(\boldsymbol{\beta})_j)}{\Gamma\left(\sum_{j=1}^{V}(\mathbf{n_k})_j+(\boldsymbol{\beta})_j\right)}\right]\cdot\left[\frac{\Gamma\left(\sum_{i=1}^{K}(\boldsymbol{\alpha})_i\right)}{\Gamma\left(N_d+\sum_{i=1}^{K}(\boldsymbol{\alpha})_i\right)}\prod_{k=1}^{K}\frac{\Gamma((\mathbf{m})_k+(\boldsymbol{\alpha})_k)}{\Gamma((\boldsymbol{\alpha})_k)}\right]
\end{aligned}
\tag{3.7}
$$

The application of LDA to only one document is not very informative. Therefore we now assume that the corpus consists of M documents. Because each document has a document-topic distribution that is independent of all other documents, the extension of equation 3.7 to the case in which there are M documents is not very difficult. The same steps as before for one document are followed.

$$
\begin{aligned}
p(\mathbf{W}=\mathbf{w},\mathbf{Z}=\mathbf{z}) &= \mathbb{E}\left[\prod_{d=1}^{M}\prod_{i=1}^{N_d}p((\mathbf{W_d})_i=(\mathbf{w_d})_i,(\mathbf{Z_d})_i=(\mathbf{z_d})_i|\boldsymbol{\Theta}_{\mathbf{d}},\boldsymbol{\Phi})\right]\\
&= \mathbb{E}\left[\prod_{d=1}^{M}\prod_{i=1}^{N_d}(\boldsymbol{\Phi}_{(\mathbf{z_d})_\mathbf{i}})_{(\mathbf{w_d})_i}\cdot(\boldsymbol{\Theta}_{\mathbf{d}})_{(\mathbf{z_d})_i}\right]\\
&= \mathbb{E}\left[\prod_{d=1}^{M}\prod_{i=1}^{N_d}(\boldsymbol{\Phi}_{(\mathbf{z_d})_\mathbf{i}})_{(\mathbf{w_d})_i}\right]\cdot\mathbb{E}\left[\prod_{d=1}^{M}\prod_{i=1}^{N_d}(\boldsymbol{\Theta}_{\mathbf{d}})_{(\mathbf{z_d})_i}\right]\\
&= \left(\prod_{k=1}^{K}\mathbb{E}\left[\prod_{j=1}^{V}(\boldsymbol{\Phi}_{\mathbf{k}})_j^{(\mathbf{n_k})_j}\right]\right)\cdot\prod_{d=1}^{M}\mathbb{E}\left[\prod_{k=1}^{K}(\boldsymbol{\Theta}_{\mathbf{d}})_k^{(\mathbf{m_d})_k}\right]
\end{aligned}
\tag{3.8}
$$

The word and topic count vector from before, n and m, are slightly changed. Now, $(\mathbf{n_k})_j$ represents the number of times we observe word-topic pair $(w,z)=(j,k)$ in the whole corpus, thus in all documents. $(\mathbf{m_d})_k$ is the frequency of topic k in document d, so this count is still on document level. Again, we can apply the formulas for the product moment of a Dirichlet dirichlet distribution and we arrive at:

$$
\begin{aligned}
p(\mathbf{W}=\mathbf{w},\mathbf{Z}=\mathbf{z}) &= \left(\prod_{k=1}^{K}\mathbb{E}\left[\prod_{j=1}^{V}(\boldsymbol{\Phi}_{\mathbf{k}})_j^{(\mathbf{n_k})_j}\right]\right)\cdot\prod_{d=1}^{M}\mathbb{E}\left[\prod_{k=1}^{K}(\boldsymbol{\Theta}_{\mathbf{d}})_k^{(\mathbf{m_d})_k}\right]\\
&= \left(\prod_{k=1}^{K}\left[\frac{\Gamma\left(\sum_{i=1}^{V}(\boldsymbol{\beta})_i\right)}{\Gamma\left(\sum_{j=1}^{V}(\mathbf{n_k})_j+(\boldsymbol{\beta})_j\right)}\prod_{j=1}^{V}\frac{\Gamma((\mathbf{n_k})_j+(\boldsymbol{\beta})_j)}{\Gamma((\boldsymbol{\beta})_j)}\right]\right)\cdot\left(\prod_{d=1}^{M}\left[\frac{\Gamma\left(\sum_{i=1}^{K}(\boldsymbol{\alpha})_i\right)}{\Gamma\left(\sum_{k=1}^{K}(\mathbf{m_d})_k+(\boldsymbol{\alpha})_k\right)}\prod_{k=1}^{K}\frac{\Gamma((\mathbf{m_d})_k+(\boldsymbol{\alpha})_k)}{\Gamma((\boldsymbol{\alpha})_k)}\right]\right)\\
&= \left(\frac{\Gamma(\sum_{j=1}^{V}(\boldsymbol{\beta})_j)}{\prod_{j=1}^{V}\Gamma((\boldsymbol{\beta})_j)}\right)^{K}\cdot\left(\frac{\Gamma(\sum_{k=1}^{K}(\boldsymbol{\alpha})_k)}{\prod_{k=1}^{K}\Gamma((\boldsymbol{\alpha})_k)}\right)^{M}\cdot\prod_{k=1}^{K}\frac{\prod_{j=1}^{V}\Gamma((\mathbf{n_k})_j+(\boldsymbol{\beta})_j)}{\Gamma(\sum_{j=1}^{V}(\mathbf{n_k})_j+(\boldsymbol{\beta})_j)}\cdot\prod_{d=1}^{M}\frac{\prod_{k=1}^{K}\Gamma((\mathbf{m_d})_k+(\boldsymbol{\alpha})_k)}{\Gamma(N_d+\sum_{k=1}^{K}(\boldsymbol{\alpha})_k)}
\end{aligned}
\tag{3.9}
$$

To obtain the distribution of all words $\mathbf{w}$ given the hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ only, we need to sum over all possible values of vector $\mathbf{z}$ which has a multivariate discrete distribution. Every topic $(\mathbf{z_d})_i$ in document d linked to word i, can take a value in $\{1,\dots,K\}$. Therefore, we need to sum over a huge set of possible values of $\mathbf{z}$. That is:

$$
p(\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta})=\sum_{\mathbf{z_i}}p(\mathbf{W}=\mathbf{w},\mathbf{Z}=\mathbf{z_i}|\boldsymbol{\alpha},\boldsymbol{\beta})
\tag{3.10}
$$

Where $\mathbf{z_i}$ is some configuration of all topic assignments in the corpus. This vector has length $\sum_{d=1}^{M}N_d$, namely the same length as the corpus. The number of possible $\mathbf{z_i}$ configurations is therefore $K^{\sum_{d=1}^{M}N_d}$. This sum will cause computational problems, therefore it is considered very challenging to compute the actual corpus probability.

3.4. Improvements and adaptations to basic LDA model

Latent Dirichlet Allocation is the simplest unsupervised topic model. Because it is applied in different scientific fields [19], there are many extensions and applications of LDA, of which the most important ones (invented between 2003 and 2017) are summarized in [19]. To give an idea of the vast area of modeling possibilities, we mention the extensions that are the most interesting for applications in opinion mining below.

LDA is a hierarchical Bayesian model, and hierarchical models can easily be extended by just adding a node to the graphical structure, which is done in the dynamic topic model [6]. This model gives information on how the average topic distribution and the word distributions per topic evolve. The model is developed for political sciences, but can also be applied to opinion mining, as the development of the customer opinion can say something about, e.g. the durability of the product or effect of a campaign might be reflected in people's views. Another model that looks at the evolution of topics over time is presented in [51].

LDA can also be adapted by changing the model parameters that reflect our prior belief about the topics. With the adaptation in [49], we can influence the topic distribution beforehand by steering documents towards one broad topic. If there is a reasonable prior belief that this will occur in the review data set, this can improve the results of the topic model.

Multi-grain LDA makes a distinction between local and global topics [44]. This distinction can be incorporated into the model by adding a layer to the hierarchical structure of LDA. In the setting of review analyses, local topics can be thought of as ratable aspects like price or ease of use. Global topics are then the types of products or brands. In this way, you can retrieve information about competitive products and improve decision-making on product development to outperform your competitors.

To quickly summarize a large set of research papers, one can use the labels or tags that are usually mentioned in an article to improve the topic model. In this LDA extension, the labels form an extra layer in the hierarchical scheme of LDA, and the highest probable words are now given per label instead of per topic. Topics and labels are considered the same. This model can, therefore, be seen as a semi-supervised topic model, as we know the topics beforehand. Although it is more useful in the scientific world, it can also be applied to review analysis, as sometimes customers give their opinion on each aspect specifically.

As previously mentioned, topics in opinion mining or review analysis are often aspects of the product about which customers give their opinion. Therefore, it is useful to make a distinction between aspect words and background words. In the topic-aspect model described in [36] this distinction is made, and the most probable words per aspect and topic (e.g., product type) are given.

In basic LDA, a bag-of-words representation is used. This means that each document consists of a set of words, whose order is ignored. Every word is an individual entity and its relation with the surrounding words is lost. Finding topics is therefore done on document level, making it difficult to link the corresponding opinion words (e.g. 'nice') with the right aspect of the product. In sentence LDA [21] this bag-of-words assumption is slightly relaxed because it is assumed that a sentence consists of only one topic.
In the same paper, the aspect and sentiment unification model (ASUM) is described. In this model, it is assumed that every document has one sentiment and tells about different topics. As a result, highly probable words per sentiment-aspect combination are given. Therefore, one can draw more detailed conclusions on the sentiment about specific topics.

The topic sentiment mixture model (TSMM) belongs to the same type of models. In this model, a distinction between background, positive and negative words is made [31]. Again, the results consist of lists with the most probable words per topic and per type of words (background, positive or negative). Note that these models, sentence LDA, ASUM and TSMM, are comparable concerning functionality. Differences can mostly be found in the order of picking sentiments or aspects. Compare the case in which customers first decide if they are positive, neutral or negative about a product and then decide on their opinion of the aspect, with the case in which the customers first decide which topics they want to mention in their review and subsequently what their sentiment is on these topics. Other similar models are the joint sentiment topic (JST) model [26], sentiment LDA [25], dependency-sentiment LDA [25], reverse joint sentiment topic model [27] and the latent aspect rating analysis (LARA) model [50].

One of the newest extensions is called part-of-speech LDA [10]. This method introduces syntactic information, that is, the function of words in a sentence, into LDA. The bag-of-words representation is let go off, and word order is incorporated. The results consist again of lists of highly probable words, but now per combination of topic and syntax class. Syntax categories are for example nouns, adjectives, adverbs or determinants, but there are many more [15].

This last model is used as inspiration for the development of a new extension of LDA that fills the needs of CQM in their ratings and review studies.

4

Inference methods for LDA

“We may at once admit that any inference from the particular to the general must be attended with some degree of uncertainty, but this is not the same as to admit that such inference cannot be absolutely rigorous, for the nature and degree of the uncertainty may itself be capable of rigorous expression.”
Sir Ronald Fisher (1890-1962)

While Latent Dirichlet Allocation is usually described as a generative process, the actual use of it is reverse. The corpus is formed by a set of reviews, whose words are observed, while the topics they belong to are unknown. The goal of LDA is to determine about which topics customers write and which topics occur more often than others. In other words, the aim is to estimate the topic-word distributions $\mathbf{\Phi_k}$ for $k \in \{1,\dots,K\}$ and the document-topic distributions $\mathbf{\Theta_d}$ for $d \in \{1,\dots,M\}$. The topic assignments $(\mathbf{Z_d})_i$ for $d \in \{1,\dots,M\}$ and $i \in \{1,\dots,N_d\}$ are merely auxiliary variables to link the document-topic distributions with the topic-word distributions.

As described before, LDA is a hierarchical Bayesian model. On the two ends of the hierarchical structure in figure 3.1, that is on $\mathbf{\Theta}$ and $\mathbf{\Phi}$, priors are imposed, representing the degree of belief in values of the document-topic distributions and the topic-word distributions respectively. The priors are probability densities with fixed parameters, respectively $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. After having observed the data, i.e., the words, posterior probabilities can be constructed. The mechanics of Bayesian statistics were explained in section 2.1.

One of the advantages of this Bayesian way of estimating a variable is that we can give extra information to the model. Expert opinions can be taken into account by choosing a prior that reflects their belief. Thus, we can select the values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ such that they correspond with our expectation on a typical document-topic distribution and topic-word distribution. If we do not have any prior knowledge, we can choose the hyperparameters to be equal to vectors with only 1’s, which is the multivariate uniform distribution and therefore is an uninformative prior.

The posterior distribution of all hidden variables $\mathbf{\Theta}$, $\mathbf{\Phi}$ and $\mathbf{Z}$ can be expressed using Bayes’ rule, where we abuse the notation of $\mathbf{\Phi}$ and $\mathbf{\Theta}$, which actually represent sets of vectors: $\mathbf{\Phi} = \left\{\boldsymbol{\phi_1},\dots,\boldsymbol{\phi_K}\right\}$ and $\mathbf{\Theta} = \{\boldsymbol{\theta_1},\dots,\boldsymbol{\theta_M}\}$. In $\mathbf{w}$, all words from all documents are collected, and in $\mathbf{z}$ all topics.

$$p(\boldsymbol{\theta},\boldsymbol{\phi},\mathbf{z}|\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{p(\boldsymbol{\theta},\boldsymbol{\phi},\mathbf{z},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta})}{p(\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta})} \tag{4.1}$$

However, we are only interested in $\mathbf{\Theta_d}$ for $d = 1,\dots,M$, and $\mathbf{\Phi_k}$ for $k = 1,\dots,K$, so we can marginalize out the topic assignments $\mathbf{z}$.

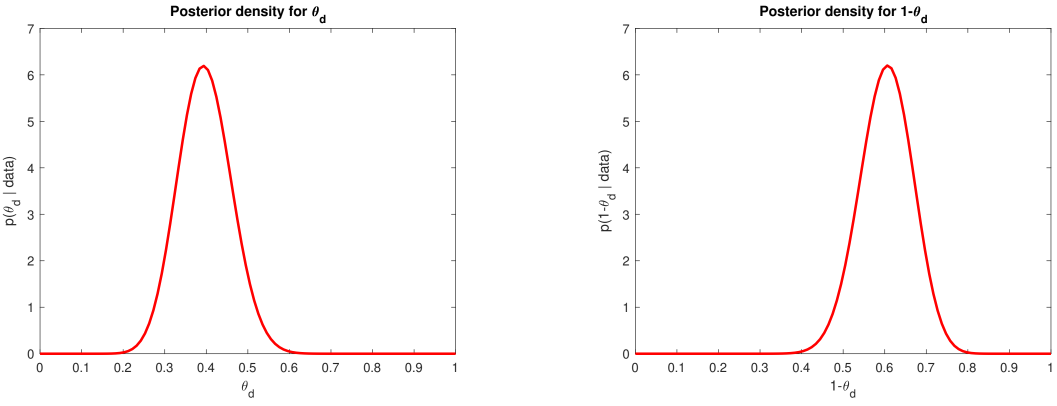
Then, the posterior becomes:

$$
\begin{aligned}
p(\boldsymbol{\theta},\boldsymbol{\phi}|\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta}) &= \frac{p(\mathbf{w}|\boldsymbol{\theta},\boldsymbol{\phi})\,p(\boldsymbol{\theta},\boldsymbol{\phi}|\boldsymbol{\alpha},\boldsymbol{\beta})}{p(\mathbf{w})} \\
&= \frac{\left[\prod_{d=1}^{M}\prod_{i=1}^{N_d} p(\widetilde{\mathbf{w}}_{\mathbf{d,i}}|\boldsymbol{\theta}_{\mathbf{d}},\boldsymbol{\phi})\right]\cdot\left[\prod_{d=1}^{M} p(\boldsymbol{\theta}_{\mathbf{d}}|\boldsymbol{\alpha})\right]\cdot\left[\prod_{k=1}^{K} p(\boldsymbol{\phi}_{\mathbf{k}}|\boldsymbol{\beta})\right]}{p(\mathbf{w})} \\
&= \frac{\left[\prod_{d=1}^{M}\prod_{i=1}^{N_d}\left(\sum_{k=1}^{K} p(\widetilde{\mathbf{w}}_{\mathbf{d,i}}|(\mathbf{z}_{\mathbf{d}})_i = k,\boldsymbol{\phi}_{\mathbf{k}})\,p((\mathbf{z}_{\mathbf{d}})_i = k|\boldsymbol{\theta}_{\mathbf{d}})\right)\right]\cdot\left[\prod_{d=1}^{M} p(\boldsymbol{\theta}_{\mathbf{d}}|\boldsymbol{\alpha})\right]\cdot\left[\prod_{k=1}^{K} p(\boldsymbol{\phi}_{\mathbf{k}}|\boldsymbol{\beta})\right]}{p(\mathbf{w})} \\
&= \frac{\left[\prod_{d=1}^{M}\prod_{i=1}^{N_d}\left(\sum_{k=1}^{K} (\boldsymbol{\phi}_{\mathbf{k}})_{(\mathbf{w}_{\mathbf{d}})_i}(\boldsymbol{\theta}_{\mathbf{d}})_k\right)\right]\cdot\left[\prod_{d=1}^{M} p(\boldsymbol{\theta}_{\mathbf{d}}|\boldsymbol{\alpha})\right]\cdot\left[\prod_{k=1}^{K} p(\boldsymbol{\phi}_{\mathbf{k}}|\boldsymbol{\beta})\right]}{p(\mathbf{w})} \\
&\propto \left[\prod_{d=1}^{M}\prod_{j=1}^{V}\left(\sum_{k=1}^{K}(\boldsymbol{\phi}_{\mathbf{k}})_j(\boldsymbol{\theta}_{\mathbf{d}})_k\right)^{n_{d,j}}\right]\cdot\left[\prod_{d=1}^{M}\prod_{k=1}^{K}(\boldsymbol{\theta}_{\mathbf{d}})_k^{(\boldsymbol{\alpha})_k-1}\right]\cdot\left[\prod_{k=1}^{K}\prod_{j=1}^{V}(\boldsymbol{\phi}_{\mathbf{k}})_j^{(\boldsymbol{\beta})_j-1}\right]
\end{aligned}
\tag{4.2}
$$

Here $n_{d,j}$ is the frequency of word j in document d.
The first term between square brackets of the right-hand side makes posterior inference difficult, due to the coupling between the topic-word distribution parameters $(\boldsymbol{\Phi}_{\mathbf{k}})_j$ and the document-topic distribution parameters $(\boldsymbol{\Theta}_{\mathbf{d}})_k$ and a summation.

Posterior inference aims to retrieve an estimation for the parameters of interest. Possible estimators in the Bayesian setting are the posterior mean or the posterior mode. We will consider both the posterior mean and posterior mode for the posterior inference on LDA. In literature, it is often immediately concluded that the posterior is intractable, see, e.g. [7], and approximation methods are applied. We do not wish to draw the same conclusion that quickly, therefore we search for cases in which 'analytical' posterior inference can be done.

As mentioned in chapter 2, there are two possibilities for estimators based on the posterior distribution: the posterior mean and the posterior mode. In the case of LDA, multiple sources (e.g. [7]) mention multimodality of the posterior distribution. Therefore, taking the posterior mean might not be wise, because it averages over the modes.
This phenomenon is shown for the simple case in which there are two topics. There is only one parameter (actually a random variable) Θ_d, namely the probability of the document belonging to the first topic. The probability of the second topic is then $1-\Theta_d$. Consider the Bayesian statistics example from chapter 2 and figures 2.1a and 2.1b. Now take a closer look at the posterior distribution:

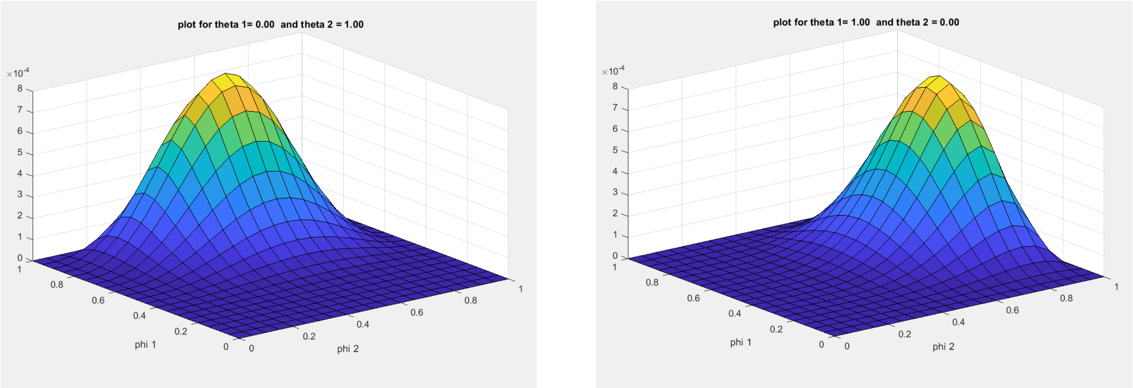


(a) Posterior density of Θ_d.

(b) Posterior density of $1-\Theta_d$.

Figure 4.1: Posterior densities of Θ_d and $1-\Theta_d$, i.e. the probabilities of respectively topic 1 and 2 for document d.

Both the posterior mean and the posterior mode are good estimators for Θ_d. Naturally, the estimator for the probability of topic 2, is then $1-\hat{\theta}_d$. Only one aspect of LDA has not been taken into account at the moment, namely the topic exchangeability. We can call topic 1, topic 2 and vice versa. There is nothing wrong with this, as the index of a topic is just a name. In that case, the two graphs in figure 4.1 interchange and the posterior mode probability of topic 1 is approximately 0.6 instead of 0.4. This topic exchangeability causes multiple modes to arise in the complete posterior, where both $\mathbf{\Theta}$ and $\mathbf{\Phi}$ are variables. The computation of the posterior mean for each Θ_d then results in a value equal to $\frac{1}{K}$ (approximately, depending on the data). The multimodality in an example with two topics, two documents, and two possible words is shown in figure 4.2.



(a) Posterior density $p(\theta_1 = 0, \theta_2 = 1, \phi_1, \phi_2)$.

(b) Posterior density $p(\theta_1 = 1, \theta_2 = 0, \phi_1, \phi_2)$.

Figure 4.2: Visualization of the posterior densities $p(\theta_1, \theta_2, \phi_1, \phi_2 | \mathbf{w_1}, \mathbf{w_2}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ with θ_1 and θ_2 fixed on the value of a mode. The maximum of the 4-dimensional posterior density is found using grid search. The two documents consists only of two possible words, for ease of notation denoted by 1 and 2. Document 1 is then '2 2 1 1 1' and document 2 consists of '1 1 1 1 1 1 2'. Hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are set to respectively 0.9 and 1.1 and are symmetric i.e. the same for all dimensions.

The two modes in figure 4.2 contain the same information and actually also show the same result. The posterior modes are $\theta_1 = 1$, $\theta_2 = 0$, $\phi_1 = 0.6$, $\phi_2 = 0.85$ and $\theta_1 = 0$, $\theta_2 = 1$, $\phi_1 = 0.85$, $\phi_2 = 0.6$, where it can be easily seen to the two topics are interchanged to get from the first mode to the second. The posterior mean values for the parameters are $\theta_1 = 0.5$, $\theta_2 = 0.5$, $\phi_1 = 0.64$, $\phi_2 = 0.64$. As a result, we see that indeed the posterior mean values for θ are $\frac{1}{K}$. Thus, in terms of topic distributions per document, the posterior mean is uninformative. It is not realistic that every document is about all topics in the same extent. Therefore, the posterior mean is not the best choice as estimator.

4.1. Posterior mean

Nevertheless, research is done on the computation of the posterior mean for LDA in any dimension. A summary of this research is given in the next section. The explanation of the possible methods of calculation of the posterior mean and their disadvantages can be found in the appendix A.3.

Although posterior mean estimation may not be very informative in terms of useful topics, Markov chain Monte Carlo methods show unusual behavior for LDA. The fact that MCMC methods do not work well for the problematic form of the posterior density results actually in good estimations of the latent variables, as will be explained in section 4.1.2.

4.1.1. Analytical determination

First of all, one might think that it is not possible to compute the posterior mean, as we know the posterior distribution only up to a constant. That is, the term $p(\mathbf{w})$ is too difficult to calculate. Blei et al. even conclude that it is the reason for intractability of the posterior [7].

In many cases in statistics, it is true that the posterior mean cannot be determined if the posterior distribution is not fully known because the proportionality constant influences the actual value of the posterior mean. However, in this case, we are lucky, as we know beforehand that our parameter vector elements need to sum to 1. That is $\sum_i (\mathbf{\Theta_d})_i = 1$ for all $d \in \{1, \dots M\}$ and $\sum_j (\mathbf{\Phi_k})_j = 1$ for all $k \in \{1, \dots K\}$. Therefore, assume we know the posterior mean up to a constant a:

$$(\hat{\boldsymbol{\theta}}_\mathbf{d})_i^{(\text{true})} = a \cdot (\hat{\boldsymbol{\theta}}_\mathbf{d})_i^{(\text{est})} \tag{4.3}$$

with *true* denoting the true posterior mean and *est* the posterior mean estimated using the posterior in equation 4.2. Then the actual posterior mean values can be determined with:

$$\begin{aligned}(\hat{\boldsymbol{\theta}}_{\mathbf{d}})_i^{(\text{true})} &= \frac{a \cdot (\hat{\boldsymbol{\theta}}_{\mathbf{d}})_i^{(\text{est})}}{\sum_{i=1}^{K} a \cdot (\hat{\boldsymbol{\theta}}_{\mathbf{d}})_i^{(\text{est})}} \\ &= \frac{(\hat{\boldsymbol{\theta}}_{\mathbf{d}})_i^{(\text{est})}}{\sum_{i=1}^{K} (\hat{\boldsymbol{\theta}}_{\mathbf{d}})_i^{(\text{est})}}\end{aligned} \tag{4.4}$$

The same procedure can be followed for posterior mean estimators for all vectors $\mathbf{\Phi_k}$, $k = 1, \ldots K$. Hence, the actual value of $p(\mathbf{w})$ is not needed for the computation of the posterior means and we can continue with the posterior in equation 4.2.

In appendix A.3 is explained how this posterior mean can be computed analytically, and if not possible, how it can be approximated. Unfortunately, the circumstances and model choices in Latent Dirichlet Allocation are such that neither analytical computation nor feasible approximation methods of the posterior mean are possible.
Therefore, it is better to focus on either the posterior mode or the posterior mean of a subspace of the domain around the posterior mode. The latter can be approximated using Markov chain Monte Carlo methods and is elaborated on in the section.

4.1.2. Markov chain Monte Carlo methods

Markov chain Monte Carlo (MCMC) methods form a collection of techniques with which samples of the posterior distribution can be obtained. If there are enough samples, the techniques result in a good approximation of the posterior distribution. With this posterior density approximation, the posterior mean can be computed. The official definition of an MCMC method is given in [46].

Definition 4.1 (MCMC method)
A Markov chain Monte Carlo method for the simulation of a distribution π is any method producing an ergodic Markov chain whose stationary distribution is π.

Remember that one can think of an ergodic Markov chain as a chain in which each state can be reached from every other state. The stationary distribution of the Markov chain is its distribution in the limit of infinitely many samples.

There are various simulation methods proposed in the literature: the Metropolis-Hastings algorithm, Gibbs sampling and collapsed Gibbs sampling, where the latter is an extension of Gibbs sampling.

Metropolis-Hastings

The Metropolis-Hastings algorithm has been invented in 1953 by among others Nicholas Metropolis. W.K. Hastings generalized his idea in 1970 to the now commonly known ‘Metropolis-Hastings’ algorithm. To summarize the idea of this algorithm, the notation of Smith and Roberts in [41] is followed.

Let $\pi(\mathbf{x}) = \pi(x_1, \ldots, x_k)$ for $x_j \in \mathbb{R}^n$, $j = 1, \ldots, k$, denote a joint density and let $\pi(x_i|\mathbf{x}_{-i})$ denote the conditional densities of x_i for $i \in \{1, \ldots, k\}$ given all other x_j's, i.e. $\mathbf{x}_{-i} = (x_j, j \neq i)$. The goal of the algorithm is to construct a Markov chain $X^1, \ldots, X^t, \ldots$ with state space Ω and equilibrium distribution $\pi(\mathbf{x})$. The state space Ω is the space of all values that $\mathbf{x}$ can take.
The Metropolis-Hastings algorithm works with transition probabilities from one state to the next, e.g. $X^t = \mathbf{x}$ to $X^{t+1} = \mathbf{y}$, for some state $\mathbf{y} \in \Omega$. These transition probabilities are denoted with the transition probability function q, such that $q(\mathbf{x}, \mathbf{y}) = \mathbb{P}(X^t = \mathbf{x}, X^{t+1} = \mathbf{y})$.

However, further randomization is applied: with some probability the new state in X^{t+1} is accepted, while with complementary probability the new state is rejected and X^{t+1} remains in the same state as X^t. Intuitively speaking, we do not jump to the next state, but remain in the state in which we already were. Formally defined, the transition probability of the Markov chain is given by function p. For ease of notation, we have left out the vector notation.

$$p(x,y)=\begin{cases} q(x,y)\alpha(x,y) & \text{if } y\neq x\\ 1-\sum_{y'\in\Omega} q(x,y')\alpha(x,y') & \text{if } y=x\end{cases} \tag{4.5}$$

With:

$$\alpha(x,y)=\begin{cases}\min\left\{\frac{\pi(y)q(y,x)}{\pi(x)q(x,y)},1\right\} & \text{if } \pi(x)q(x,y)>0\\ 1 & \text{if } \pi(x)q(x,y)=0\end{cases} \tag{4.6}$$

Note that the so-called detailed balance is complied to: $\pi(x)p(x,y)=\pi(y)p(y,x)$, which is necessary for the Markov chain to be irreducible and aperiodic, ergo ergodic. With this condition, $\pi(x)$ will be the equilibrium distribution of the Markov chain, as desired [41]. Furthermore, because in the computation of $\alpha(x,y)$ the joint density π arises both in the numerator and the denominator, it is only needed to know π up to a proportionality constant.

For the transition probability function q there are several possibilities to choose from [46]. If π is a continuous density, one can take for example a random walk distribution for q. Another possibility is taking $q(x,\cdot)=f(\cdot)$, such that the transition probability from state x is independent of x. It is useful to take for f a probability density that resembles the target function π, but is tractable, in contrast to π. More possibilities for q can be found in [46].

Because in the case of LDA, there are many parameters, resulting in a high dimensionality, it is difficult to find good proposal densities q [46]. Therefore, we will not dive more deeply into the Metropolis-Hastings algorithm and its properties, but rather resort to Gibbs sampling.

Gibbs sampling

Often applied to (variants of) LDA is Gibbs sampling, which is actually a special case of the one-at-a-time version of the Metropolis-Hastings algorithm.

In *one-at-a-time Metropolis-Hastings*, only one component of the vector $\mathbf{x}$ is sampled at a time, as the name suggests. For each x_i, $i\in\{1,\dots,k\}$, there is a specific transition probability function q_i. First, consider a two-dimensional example of sampling $\pi(x_1,x_2)$, where $\mathbf{x}=(x_1,x_2)$ denotes the current state, and $\mathbf{y}=(y_1,y_2)$ the next state. Thus, suppose we have (x_1,x_2), the Markov chain evolves using the following steps [46]:

1. Draw $y_1\sim q_1(x_1,y_1|x_2)$ where conditioning on x_2 means that you keep the value of x_2 fixed in this step.
2. Accept the transition with probability α_1:

$$\alpha_1=\min\left\{\frac{\pi(y_1,x_2)q_1(y_1,x_1|x_2)}{\pi(x_1,x_2)q_1(x_1,y_1|x_2)},1\right\} \tag{4.7}$$

   else $y_1=x_1$ (no jump).

3. Draw $y_2\sim q_2(x_2,y_2|y_1)$ where conditioning on y_1 means that you keep the obtained value of the first component (y_1) from the current/next index in the Markov chain fixed.
4. Accept the transition with probability α_2:

$$\alpha_2=\min\left\{\frac{\pi(y_1,y_2)q_2(y_2,x_2|y_1)}{\pi(y_1,x_2)q_2(x_2,y_2|y_1)},1\right\} \tag{4.8}$$

   else $y_2=x_2$ (no jump).

The *Gibbs sampler algorithm* then arises when you use the following transition probability functions q_i, again for the 2-dimensional example:

$$\begin{aligned} q_1(x_1,y_1|x_2)&=\pi(y_1|x_2)\\ q_2(x_2,y_2|y_1)&=\pi(y_2|x_1)\end{aligned} \tag{4.9}$$

That is, when the transition probability functions q_i are defined as the conditional distributions derived from the target distribution π and only one component of the random vector is sampled at a time, the Metropolis-Hastings method becomes Gibbs sampling. Note that with these functions q_i, the acceptance probabilities α_i will always be 1, therefore, you always draw the next state in the Markov chain from the conditional distribution.

Gibbs sampling is only applicable when the conditional distributions are of a known form. Therefore, Bayesian statisticians usually use conjugate priors, such that the posterior distribution is from the same family of distributions as the prior. Then, one can easily sample from conditional distributions. If the posterior distribution is not a known density, it is very challenging to obtain samples, which is why in those cases Metropolis-Hastings is used.

One of the main advantages of Gibbs sampling or Markov chain Monte Carlo methods in general is that they converge under relatively weak assumptions. In [42], these assumptions are explained. Define $K(\mathbf{x},\mathbf{y})$ to be the transition kernel of the Markov chain, that is, for the two-dimensional example above [42]:

$$K(\mathbf{x},\mathbf{y}) = \pi(y_1|x_2)\cdot\pi(y_2|y_1) \tag{4.10}$$

provided that $\int \pi(y_1,x_2)\,dy_1 > 0$ and $\int \pi(y_1,y_2)\,dy_2 > 0$, otherwise in determining the conditional densities, we would divide by 0. If one of these conditions in not satisfied, $K(\mathbf{x},\mathbf{y}) = 0$ by definition. The kernel $K(\mathbf{x},\mathbf{y})$ maps from $D\times D$ to $\mathbb{R}^2$, where $D = \{\mathbf{x}\in\Omega, \pi(x) > 0\}$ with Ω the state space of $\mathbf{x}$, in this section chosen to be $\mathbb{R}^2$.
Smith and Roberts state convergence of the Gibbs sampler in the following theorem:

Theorem 4.1 (Convergence Gibbs sampler)
If K is π-irreducible and aperiodic, then for all $\mathbf{x}\in D$:

1. $\int_\Omega |K^{(t)}(\mathbf{x},\mathbf{y}) - \pi(\mathbf{y})|d\mathbf{y} \to 0$ *for* $t\to\infty$.
2. *for real-valued, π-integrable function f,*

$$t^{-1}\left(f(X^1)+\dots f(X^t)\right) \to \int_\Omega f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$$

almost surely for $t\to\infty$.

Here t is the number of samples in the Markov chain. $K^{(t)}$ is the kernel describing t iterations, more thoroughly explained in [42]. From the theorem, we can conclude that the kernel K converges to π in L^1. Furthermore, the sample mean of all samples in the Markov chain (corrected for the initial, transient phase, also called the burn-in period) converges to the actual mean of π almost surely. Hence, in Gibbs sampling, the sample mean is used as an approximation for the posterior mean.
To use theorem 4.1, we need to show that the kernel is π-irreducible and aperiodic. Smith and Roberts give simple conditions for the standard Gibbs sampling formulas to satisfy these two assumptions in [42]. They explain that it is hard to find settings for Gibbs sampling in which these two assumptions are not satisfied. Therefore, we expect Gibbs sampling to be a reliable method to obtain posterior mean estimate for the latent random variables of interest in LDA.

Example Gibbs sampling To illustrate the algorithm of Gibbs sampling, a 2-dimensional example is worked out, which is, in fact, the 2-dimensional basis of LDA. Therefore, the link to higher dimensional is easy to make, as the hierarchical scheme of LDA has already been explained.
Consider a group of students that makes a test. The test consists of N questions, and each student is assumed to have a probability of Θ_i to give the right answer to a question. There are n students, and the only observed variables are $X_1,\dots X_n$, the number of questions that each student has answered correctly. Furthermore, we assume that each student works independently and all questions are equally difficult, so $X_i \sim \text{Binomial}(\Theta_i, N)$ with no correlation between the X_i. The probability of giving the right answer is a priori Beta distributed i.e. $\Theta_i \sim \text{Beta}(a,b)$ $i.i.d.$ Then the posterior distribution of $\mathbf{\Theta}|\mathbf{X}$ can be derived.

$$\begin{aligned}
p(\boldsymbol{\theta}|\mathbf{x},a,b) &= \frac{p(\boldsymbol{\theta},\mathbf{x}|a,b)}{p(\mathbf{x}|a,b)} \\
&\propto p(\mathbf{x}|\boldsymbol{\theta},a,b)p(\boldsymbol{\theta}|a,b) \\
&= \prod_{i=1}^{n}\binom{n}{x_i}\theta_i^{x_i}(1-\theta_i)^{N-x_i}\cdot\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\theta_i^{a-1}(1-\theta_i)^{b-1} \\
&\propto \prod_{i=1}^{n}\theta_i^{x_i+a-1}(1-\theta_i)^{N-x_i+b-1}
\end{aligned} \tag{4.11}$$

Because Θ_i is independent of all other Θ_j for $j \neq i$, we can easily see that:

$$\begin{aligned} p(\theta_i|\theta_1,\theta_{i-1},\theta_{i+1},\theta_n,\mathbf{x},a,b) &\propto \theta_i^{x_1+a-1}(1-\theta_i)^{N-x_i+b-1} \\ \Rightarrow \Theta_i|x_i,a,b &\sim \text{Beta}(x_i+a,N-x_i+b) \end{aligned} \tag{4.12}$$

Note that the Beta distribution is a conjugate prior to the Binomial distribution. The Gibbs sampling procedure becomes:

- Draw θ_1 from Beta$(x_1+a,N-x_1+b)$
- ...
- Draw θ_n from Beta$(x_n+a,N-x_n+b)$

When this procedure is executed sufficiently often, the samples for each Θ_i will approximate the posterior distribution $p(\theta_i|x_i,a,b)$ with which then the value of Θ_i can be estimated via the posterior mode or posterior mean.

Gibbs sampling for LDA In Latent Dirichlet Allocation, many latent variables need to be retrieved using Gibbs sampling. The observed data are the words $\mathbf{w}$, where $\mathbf{w}$ has a length of $\sum_{d=1}^{M} N_d$. The fixed parameters, also called hyperparameters are $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The unknown parameters are $\mathbf{\Theta_d}$ for $d \in \{1,\dots,M\}$, $\mathbf{\Phi_k}$ for $k \in \{1,\dots,K\}$ and $\mathbf{Z}$, which has the same length as $\mathbf{w}$. This means that there are $M+K+1$ vectors to be estimated that have $M \cdot K + K \cdot V + \left(\sum_{d=1}^{M} N_d\right)$ components together. To give an idea; assume we have 10,000 reviews with each 100 words. We assume that 20 topics are written about. In total, there are 1000 words in the vocabulary. This means that we have to estimate $10,000 \cdot 20 + 20 \cdot 1000 + 10,000 \cdot 100 = 320,000$ components. Luckily, it is possible to sample the vectors $\mathbf{\Theta_d}$ and $\mathbf{\Phi_k}$ at once, and no component-wise inference needs to be done. The conditional distributions are derived as follows.

A single document d from the set $\{1,\dots,M\}$ can be considered, because in LDA, it is assumed that each document is generated independently. The full conditional distribution of $\mathbf{\Theta_d}$ is derived.

$$\begin{aligned} p(\boldsymbol{\theta}_\mathbf{d}|\boldsymbol{\theta}_\mathbf{1},\dots,\boldsymbol{\theta}_\mathbf{d-1},\boldsymbol{\theta}_\mathbf{d+1},\dots,\boldsymbol{\theta}_\mathbf{M},\boldsymbol{\phi}_\mathbf{1},\dots,\boldsymbol{\phi}_\mathbf{K},\mathbf{z},\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta}) &= \frac{p(\boldsymbol{\theta}_\mathbf{1},\dots,\boldsymbol{\theta}_\mathbf{M},\boldsymbol{\phi}_\mathbf{1},\dots,\boldsymbol{\phi}_\mathbf{K},\mathbf{z},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta})}{p(\boldsymbol{\theta}_\mathbf{1},\dots,\boldsymbol{\theta}_\mathbf{d-1},\boldsymbol{\theta}_\mathbf{d+1},\dots,\boldsymbol{\theta}_\mathbf{M},\boldsymbol{\phi}_\mathbf{1},\dots,\boldsymbol{\phi}_\mathbf{K},\mathbf{z},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta})} \\ &\propto p(\boldsymbol{\theta}_\mathbf{1},\dots,\boldsymbol{\theta}_\mathbf{M},\boldsymbol{\phi}_\mathbf{1},\dots,\boldsymbol{\phi}_\mathbf{K},\mathbf{z},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta}) \\ &\propto p(\boldsymbol{\theta}_\mathbf{d},\boldsymbol{\phi}_\mathbf{1},\dots,\boldsymbol{\phi}_\mathbf{K},\mathbf{z}_\mathbf{d},\mathbf{w}_\mathbf{d}|\boldsymbol{\alpha},\boldsymbol{\beta}) \\ &= \left(\prod_{i=1}^{N_d} p(\widetilde{\mathbf{w}}_i|\widetilde{\mathbf{z}}_i,\boldsymbol{\theta}_\mathbf{d},\boldsymbol{\phi}_\mathbf{1},\dots,\boldsymbol{\phi}_\mathbf{K},\boldsymbol{\alpha},\boldsymbol{\beta})\, p(\widetilde{\mathbf{z}}_i|\boldsymbol{\theta}_\mathbf{d},\boldsymbol{\phi}_\mathbf{1},\dots,\boldsymbol{\phi}_\mathbf{K},\boldsymbol{\alpha},\boldsymbol{\beta})\right) \cdot \left(\prod_{k=1}^{K} p(\boldsymbol{\phi}_\mathbf{k}|\boldsymbol{\beta})\right) \cdot p(\boldsymbol{\theta}_\mathbf{d}|\boldsymbol{\alpha}) \\ &= \left(\prod_{i=1}^{N_d} p(\widetilde{\mathbf{w}}_i|\widetilde{\mathbf{z}}_i,\boldsymbol{\phi}_\mathbf{1},\dots,\boldsymbol{\phi}_\mathbf{K})\, p(\widetilde{\mathbf{z}}_i|\boldsymbol{\theta}_\mathbf{d})\right) \cdot \left(\prod_{k=1}^{K} p(\boldsymbol{\phi}_\mathbf{k}|\boldsymbol{\beta})\right) \cdot p(\boldsymbol{\theta}_\mathbf{d}|\boldsymbol{\alpha}) \\ &\propto \left(\prod_{i=1}^{N_d} p(\widetilde{\mathbf{z}}_i|\boldsymbol{\theta}_\mathbf{d})\right) \cdot p(\boldsymbol{\theta}_\mathbf{d}|\boldsymbol{\alpha}) \\ &\propto \left(\prod_{i=1}^{N_d}\prod_{k=1}^{K} (\boldsymbol{\theta}_\mathbf{d})_k^{(\widetilde{\mathbf{z}}_\mathbf{i})_k}\right) \cdot \prod_{k=1}^{K} (\boldsymbol{\theta}_\mathbf{d})_k^{(\boldsymbol{\alpha})_k-1} \\ &= \prod_{k=1}^{K} (\boldsymbol{\theta}_\mathbf{d})_k^{(\mathbf{m}_\mathbf{d})_k+(\boldsymbol{\alpha})_k-1} \end{aligned} \tag{4.13}$$

Here we define $m_{d,k}$ as the number of times a word in document d is assigned to topic k. The expression in 4.13 can be recognized as a Dirichlet distribution:

$$\left(\mathbf{\Theta_d}|\boldsymbol{\theta}_\mathbf{1},\dots,\boldsymbol{\theta}_\mathbf{d-1},\boldsymbol{\theta}_\mathbf{d+1},\dots,\boldsymbol{\theta}_\mathbf{M},\boldsymbol{\phi}_\mathbf{1},\dots,\boldsymbol{\phi}_\mathbf{K},\mathbf{z},\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta}\right) \sim \text{Dirichlet}(\mathbf{m_d}+\boldsymbol{\alpha}) \tag{4.14}$$

With $\mathbf{m_d}$ the vector of topic frequencies $(\mathbf{m_d})_k$ for $k = 1,\dots,K$.

Next, the full conditional of $\mathbf{\Phi_t}$ for some $t \in \{1,\dots,K\}$ is derived.

$$\begin{aligned} p(\boldsymbol{\phi}_\mathbf{t}|\boldsymbol{\theta}_\mathbf{1},\dots,\boldsymbol{\theta}_\mathbf{M},\boldsymbol{\phi}_\mathbf{1},\dots,\boldsymbol{\phi}_\mathbf{t-1},\boldsymbol{\phi}_\mathbf{t+1},\dots,\boldsymbol{\phi}_\mathbf{K},\mathbf{z},\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta}) &\propto p(\boldsymbol{\theta}_\mathbf{1},\dots,\boldsymbol{\theta}_\mathbf{M},\boldsymbol{\phi}_\mathbf{1},\dots,\boldsymbol{\phi}_\mathbf{K},\mathbf{z},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta}) \\ &= \prod_{d=1}^{M}\left[\prod_{i=1}^{N_d} p(\widetilde{\mathbf{w}}_\mathbf{d,i}|\widetilde{\mathbf{z}}_\mathbf{d,i},\boldsymbol{\phi}_{(\mathbf{z_d})_\mathbf{i}})\,p(\widetilde{\mathbf{z}}_\mathbf{d,i}|\boldsymbol{\theta}_\mathbf{d})\,p(\boldsymbol{\theta}_\mathbf{d}|\boldsymbol{\alpha})\right]\cdot\prod_{k=1}^{K} p(\boldsymbol{\phi}_\mathbf{k}|\boldsymbol{\beta}) \\ &\propto \left[\prod_{d=1}^{M}\prod_{i=1}^{N_d} p(\widetilde{\mathbf{w}}_\mathbf{d,i}|\widetilde{\mathbf{z}}_\mathbf{d,i},\boldsymbol{\phi}_{(\mathbf{z_d})_\mathbf{i}})\right] p(\boldsymbol{\phi}_\mathbf{t}|\boldsymbol{\beta}) \\ &\propto \left[\prod_{d=1}^{M}\prod_{i=1}^{N_d}\prod_{j=1}^{V} (\boldsymbol{\phi}_{(\mathbf{z_d})_\mathbf{i}})_j^{(\widetilde{\mathbf{w}}_\mathbf{d,i})_j}\right]\prod_{j=1}^{V}(\boldsymbol{\phi}_\mathbf{t})_j^{(\boldsymbol{\beta})_j-1} \\ &\propto \prod_{j=1}^{V}(\boldsymbol{\phi}_\mathbf{t})_j^{(\mathbf{n_t})_j+(\boldsymbol{\beta})_j-1} \end{aligned} \tag{4.15}$$

where again $(\mathbf{n_t})_j$ represents the number of times word j is assigned to topic t in the whole corpus. Similarly as for $\mathbf{\Theta_d}$, the resulting conditional distribution is a Dirichlet:

$$\left(\mathbf{\Phi_t}|\boldsymbol{\theta}_\mathbf{1},\dots,\boldsymbol{\theta}_\mathbf{M},\boldsymbol{\phi}_\mathbf{1},\dots,\boldsymbol{\phi}_\mathbf{t-1},\boldsymbol{\phi}_\mathbf{t+1},\dots,\boldsymbol{\phi}_\mathbf{K},\mathbf{z},\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta}\right) \sim \text{Dirichlet}(\mathbf{n_t}+\boldsymbol{\beta}) \tag{4.16}$$

Lastly, the topics corresponding to each word in each document need to be sampled conditional on all other variables:

$$\begin{aligned} p((\mathbf{z_d})_i|\boldsymbol{\theta}_\mathbf{1},\dots,\boldsymbol{\theta}_\mathbf{M},\boldsymbol{\phi}_\mathbf{1},\dots,\boldsymbol{\phi}_\mathbf{K},\mathbf{z}_{-(d,i)},\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta}) &= p(\widetilde{\mathbf{z}}_\mathbf{d,i}|\boldsymbol{\theta}_\mathbf{1},\dots,\boldsymbol{\theta}_\mathbf{M},\boldsymbol{\phi}_\mathbf{1},\dots,\boldsymbol{\phi}_\mathbf{K},\mathbf{z}_{-(d,i)},\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta}) \\ &\propto p(\boldsymbol{\theta}_\mathbf{1},\dots,\boldsymbol{\theta}_\mathbf{M},\boldsymbol{\phi}_\mathbf{1},\dots,\boldsymbol{\phi}_\mathbf{K},\mathbf{z},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta}) \\ &= \prod_{d'=1}^{M}\left[\prod_{i'=1}^{N_{d'}} p(\widetilde{\mathbf{w}}_{d',i'}|\widetilde{\mathbf{z}}_{d',i'},\boldsymbol{\phi}_{\mathbf{z}_{\mathbf{d}',\mathbf{i}'}})\,p(\widetilde{\mathbf{z}}_{d',i'}|\boldsymbol{\theta}_\mathbf{d'})\,p(\boldsymbol{\theta}_{\mathbf{d}'}|\boldsymbol{\alpha})\right]\cdot\prod_{k=1}^{K} p(\boldsymbol{\phi}_\mathbf{k}|\boldsymbol{\beta}) \\ &\propto p(\widetilde{\mathbf{w}}_\mathbf{d,i}|\widetilde{\mathbf{z}}_\mathbf{d,i},\boldsymbol{\phi}_{(\mathbf{z_d})_\mathbf{i}})\,p(\widetilde{\mathbf{z}}_\mathbf{d,i}|\boldsymbol{\theta}_\mathbf{d}) \\ &\propto \prod_{k=1}^{K}(\boldsymbol{\theta}_\mathbf{d})_k^{(\widetilde{\mathbf{z}}_\mathbf{d,i})_k}\cdot\prod_{j=1}^{V}(\boldsymbol{\phi}_{(\mathbf{z_d})_\mathbf{i}})_k^{(\widetilde{\mathbf{w}}_\mathbf{d,i})_j} \\ &= (\boldsymbol{\theta}_\mathbf{d})_{(\mathbf{z_d})_i}\cdot(\boldsymbol{\phi}_{(\mathbf{z_d})_\mathbf{i}})_{(\mathbf{w_d})_i} \end{aligned} \tag{4.17}$$

Here we used the notation $\mathbf{z}_{-(d,i)}$ for the vector containing all topics minus the topic corresponding to word i from document d.

From expression 4.17, we can conclude that:

$$\left(\widetilde{\mathbf{Z}}_\mathbf{d,i}|\boldsymbol{\theta}_\mathbf{1},\dots,\boldsymbol{\theta}_\mathbf{M},\boldsymbol{\phi}_\mathbf{1},\dots,\boldsymbol{\phi}_\mathbf{K},\mathbf{z}_{-(d,i)},\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta}\right) \sim \text{Multinomial}((\boldsymbol{\theta}_\mathbf{d})_1\cdot(\boldsymbol{\phi}_\mathbf{1})_{(\mathbf{w_d})_i},\dots,(\boldsymbol{\theta}_\mathbf{d})_K\cdot(\boldsymbol{\phi}_\mathbf{K})_{(\mathbf{w_d})_i}) \tag{4.18}$$

In this distribution, the role of conditioning on word $(\mathbf{w_d})_i$ is clearly visible: the larger the probability of word $(\mathbf{w_d})_i$ in a topic word distribution $\mathbf{\Phi_k}$ for some topic k, the larger the probability that $(\mathbf{z_d})_i = k$. Vice versa, if the probability of $(\mathbf{w_d})_i$ is very small for, say, topic 1, the chance that $(\mathbf{z_d})_i = 1$ will be null.

The Gibbs sampling algorithm (see algorithm 1 on the next page) has excellent intuitive properties. In LDA, we only observe the words in all documents, and we fix hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Based on just this data, we want to retrieve the per document topic distribution, the per topic word distribution, and the topic assigned to each word. Initially, to all parameters is assigned a (randomly drawn) value, and iteratively they are adapted by plugging in information. That is, we update the document-topic probabilities by looking at frequencies of the topics in a document. The co-occurrence of words and topics influence the topic-word probabilities. More mathematically, the assigned topics to words are determined by a multinomial distribution that retrieves its parameters from a combination of the topic probability itself with the probability of the observed word given a particular topic. In this way, both the observed words and the hyperparameters influence the latent variables in each step, until enough samples of the conditional distributions are obtained to give a meaningful estimate of the hidden parameters.

However, the multimodality of the posterior density makes it difficult for the Gibbs sampler to 'walk' over the entire domain. The convergence result from the beginning of this section is only valid if the Gibbs sampler can

Algorithm 1 Gibbs Sampling for LDA

```
Initialize $\boldsymbol{\theta}_\mathbf{1},\dots,\boldsymbol{\theta}_\mathbf{M},\boldsymbol{\phi}_\mathbf{1},\dots,\boldsymbol{\phi}_\mathbf{K},\mathbf{z}$
Compute initial frequencies $(\mathbf{m_d})_k$ (for $d = 1$ to $M$, $k = 1$ to $K$) and $(\mathbf{n_k})_j$ (for $k = 1$ to $K$, $j = 1$ to $V$)
Fix $N_{iter}$ for maximum number of iterations
for $iter = 1$ to $N_{iter}$ do                                  ▷ Sample $N_{iter}$ times
    for $d = 1$ to $M$ do                                        ▷ Iterate over documents
        Draw $\mathbf{\Theta_d}$ from Dirichlet($\mathbf{m_d} + \boldsymbol{\alpha}$)
        for $i = 1$ to $N_d$ do                                  ▷ Iterate over words
            Draw $\widetilde{\mathbf{Z}}_{\mathbf{d,i}}$ from Multinomial($(\boldsymbol{\theta}_\mathbf{d})_1 \cdot (\boldsymbol{\phi}_\mathbf{1})_{(\mathbf{w_d})_i}, \dots, (\boldsymbol{\theta}_\mathbf{d})_K \cdot (\boldsymbol{\phi}_\mathbf{K})_{(\mathbf{w_d})_i}$)
        end for
    end for
    for $k = 1$ to $K$ do                                        ▷ Iterate over topics
        Draw $\mathbf{\Phi_k}$ from Dirichlet($\mathbf{n_k} + \boldsymbol{\beta}$)
    end for
    Update all frequencies $(\mathbf{m_d})_k$ and $(\mathbf{n_k})_j$
end for
Compute posterior estimates of variables $\mathbf{\Theta_1},\dots,\mathbf{\Theta_M},\mathbf{\Phi_1},\dots,\mathbf{\Phi_K},\mathbf{z}$ using the $N_{iter}$ samples from their posterior distributions
```

reach every value of the domain of each latent variable. An analogy can be made with two mountain tops with a ravine in between. When you, the Gibbs sampler state, are standing on one mountain, it is possible to get to the other mountain, because you cannot cross the ravine. Maybe it is possible to jump over, but the chance that you will survive is null. With probabilities between the posterior modes being very low, and the latent variables being dependent on each other, the Gibbs sampler will not move from one posterior mode to another, and therefore the estimated posterior density from the samples does not converge to the true posterior density. This might seem unfortunate, but in the application of Gibbs sampling to LDA, it helps. We are not interested in the entire posterior mean, because we already know that it will average over all possible topic permutations, resulting in $\frac{1}{K}$ for each document-topic probability. Nevertheless, because the Gibbs sampler cannot move from one topic permutation (a hill in the posterior density) to another, or it only does that with a very small probability, the posterior mean based on the Gibbs samples is the posterior mean of the samples lying around one posterior mode for some topic permutations. For this reason, the estimations found by Gibbs sampling are good estimates for $\mathbf{\Theta}$ and $\mathbf{\Phi}$, the document-topic and the topic-word distributions.

Unfortunately, there is still one downside of this procedure. As mentioned before, there are $M \cdot K + K \cdot V + \left(\sum_{d=1}^{M} N_d\right)$ latent variables and only $K + V + \sum_{d=1}^{M} N_d$ fixed/observed variables. Therefore, it is challenging to do inference and dimension reduction techniques come in place.

Collapsed Gibbs sampling

Two of the possible dimension reduction techniques are grouping and collapsing [28]. In grouping, one samples multiple parameters at a time, still using the conditional distribution on all other parameters. In a 3-dimensional example you can sample for example (x_1, x_2) conditional on x_3. Note that this technique is already secretly applied in the aforementioned Gibbs sampling method, because we sample a whole vector e.g. $\mathbf{\Theta_d}$ at once. This was done because the joint distribution of all components of, e.g., $\mathbf{\Theta_d}$ conditional on all other parameters in the model is well known. Collapsing is a technique on which the focus of this section will be. In collapsing, one variable is integrated out and only sampled after all Gibbs iterations. Looking at the simple 3-dimensional example, you can for example integrate out x_3, such that you iteratively sample x_1 from $p(x_1|x_2)$ and x_2 from $p(x_2|x_1)$. After this Gibbs sampling procedure is finished and convergence is attained, x_3 comes back into play and is sampled from $p(x_3|x_1,x_2)$. [28]

Collapsed Gibbs sampling produces results in fewer steps because one integrates out Θ and Φ, such that only the latent variables $\widetilde{\mathbf{Z}}_{\mathbf{d,i}}$ for $d = 1,\dots,M$ and $i = 1,\dots,N_d$ need to be sampled from their conditional distribution. After a certain number of iterations, many samples from the conditional posterior distributions are available, so each $\widetilde{\mathbf{Z}}_{\mathbf{d,i}}$ can be estimated. These estimates are then used to sample $\mathbf{\Theta}$ and $\mathbf{\Phi}$ from their conditional distributions. We already know from the Gibbs sampling procedure that these are Dirichlet distributed, of which we know the means. Thus, it is possible to compute the posterior means of all $\mathbf{\Theta}$ and $\mathbf{\Phi}$ directly.

The posterior distribution of $\widetilde{\mathbf{Z}}_{\mathbf{d,i}}$ conditional on all other variables (with Φ and Θ integrated out), i.e. $p(\widetilde{\mathbf{z}}_{\mathbf{d,i}}|\mathbf{w},\mathbf{z}_{-(d,i)},\boldsymbol{\alpha},\boldsymbol{\beta})$, can be derived using the joint distribution for all words and topics from equation 3.9. Let us consider the topic corresponding to word i in document d. Its conditional distribution $p(\widetilde{\mathbf{z}}_{\mathbf{d,i}}|\mathbf{w},\mathbf{z}_{-(d,i)},\boldsymbol{\alpha},\boldsymbol{\beta})$ can be expressed as follows.

$$\begin{aligned}
p((\mathbf{z_d})_i|\mathbf{w},\mathbf{z}_{-(d,i)},\boldsymbol{\alpha},\boldsymbol{\beta}) &= \frac{p(\mathbf{w},\mathbf{z}_{-(d,i)},(\mathbf{z_d})_i|\boldsymbol{\alpha},\boldsymbol{\beta}))}{p(\mathbf{w},\mathbf{z}_{-(d,i)}|\boldsymbol{\alpha},\boldsymbol{\beta})} \\
&= \frac{p(\mathbf{w},\mathbf{z}|\boldsymbol{\alpha},\boldsymbol{\beta}))}{p(\mathbf{w},\mathbf{z}_{-(d,i)}|\boldsymbol{\alpha},\boldsymbol{\beta})} \\
&\propto \frac{\left(\frac{\Gamma(\sum_{j=1}^{V}(\boldsymbol{\beta})_j)}{\prod_{j=1}^{V}\Gamma((\boldsymbol{\beta})_j)}\right)^{K}\cdot\left(\frac{\Gamma(\sum_{k=1}^{K}(\boldsymbol{\alpha})_k)}{\prod_{k=1}^{K}\Gamma((\boldsymbol{\alpha})_k)}\right)^{M}\cdot\frac{\prod_{j=1}^{V}\Gamma((\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_j+(\boldsymbol{\beta})_j)}{\Gamma(\sum_{j=1}^{V}((\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_j+(\boldsymbol{\beta})_j)}\cdot\frac{\Gamma((\mathbf{m_d})_{(\mathbf{z_d})_i}+(\boldsymbol{\alpha})_{(\mathbf{z_d})_i})}{\Gamma(\sum_{k=1}^{K}((\mathbf{m_d})_k+(\boldsymbol{\alpha})_k))}}{\left(\frac{\Gamma(\sum_{j=1}^{V}(\boldsymbol{\beta})_j)}{\prod_{j=1}^{V}\Gamma((\boldsymbol{\beta})_j)}\right)^{K}\cdot\left(\frac{\Gamma(\sum_{k=1}^{K}(\boldsymbol{\alpha})_k)}{\prod_{k=1}^{K}\Gamma((\boldsymbol{\alpha})_k)}\right)^{M}\cdot\frac{\prod_{j=1}^{V}\Gamma((\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_j^{-(d,i)}+(\boldsymbol{\beta})_j)}{\Gamma(\sum_{j=1}^{V}((\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_j^{-(d,i)}+(\boldsymbol{\beta})_j)}\cdot\frac{\Gamma((\mathbf{m_d})_{(\mathbf{z_d})_i}^{-(d,i)}+\alpha_{(\mathbf{z_d})_i})}{\Gamma(\sum_{k=1}^{K}((\mathbf{m_d})_k^{-(d,i)}+(\boldsymbol{\alpha})_k))}} \\
&= \frac{\frac{\prod_{j=1}^{V}\Gamma((\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_j+(\boldsymbol{\beta})_j)}{\Gamma(\sum_{j=1}^{V}((\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_j+(\boldsymbol{\beta})_j)}\cdot\frac{\Gamma((\mathbf{m_d})_{(\mathbf{z_d})_i}+(\boldsymbol{\alpha})_{(\mathbf{z_d})_i})}{\Gamma(\sum_{k=1}^{K}(\mathbf{m_d})_k+(\boldsymbol{\alpha})_k)}}{\frac{\prod_{j=1}^{V}\Gamma((\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_j^{-(d,i)}+(\boldsymbol{\beta})_j}{\Gamma(\sum_{j=1}^{V}(\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_j^{-(d,i)}+(\boldsymbol{\beta})_j)}\cdot\frac{\Gamma(\mathbf{m_d})_{(\mathbf{z_d})_i}^{-(d,i)}+(\boldsymbol{\alpha})_{(\mathbf{z_d})_i})}{\Gamma(\sum_{k=1}^{K}((\mathbf{m_d})_k^{-(d,i)}+(\boldsymbol{\alpha})_k))}} \\
&= \frac{\prod_{j=1}^{V}\Gamma((\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_j+(\boldsymbol{\beta})_j)\cdot\Gamma(\sum_{j=1}^{V}((\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_j^{-(d,i)}+(\boldsymbol{\beta})_j)\cdot\Gamma((\mathbf{m_d})_{(\mathbf{z_d})_i}+(\boldsymbol{\alpha})_{(\mathbf{z_d})_i})\cdot\Gamma(\sum_{k=1}^{K}(\mathbf{m_d})_k^{-(d,i)}+(\boldsymbol{\alpha})_k)}{\prod_{j=1}^{V}\Gamma((\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_j^{-(d,i)}+(\boldsymbol{\beta})_j)\cdot\Gamma(\sum_{j=1}^{V}(\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_j+(\boldsymbol{\beta})_j)\cdot\Gamma((\mathbf{m_d})_{(\mathbf{z_d})_i}^{-(d,i)}+(\boldsymbol{\alpha})_{(\mathbf{z_d})_i})\cdot\Gamma(\sum_{k=1}^{K}(\mathbf{m_d})_k+(\boldsymbol{\alpha})_k)} \\
&= \frac{(\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_{(\mathbf{w_d})_i}^{-(d,i)}+(\boldsymbol{\beta})_{(\mathbf{w_d})_i}}{\sum_{j=1}^{V}((\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_j^{-(d,i)}+(\boldsymbol{\beta})_j}\cdot\frac{(\mathbf{m_d})_{(\mathbf{z_d})_i}^{-(d,i)}+\alpha_{(\mathbf{z_d})_i}}{\sum_{k=1}^{K}((\mathbf{m_d})_k^{-(d,i)}+(\boldsymbol{\alpha})_k)} \\
&= \frac{(\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_{(\mathbf{w_d})_i}^{-(d,i)}+(\boldsymbol{\beta})_{(\mathbf{w_d})_i}}{\sum_{j=1}^{V}((\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_j^{-(d,i)}+(\boldsymbol{\beta})_j}\cdot\frac{(\mathbf{m_d})_{(\mathbf{z_d})_i}^{-(d,i)}+\alpha_{(\mathbf{z_d})_i}}{N_d-1+\sum_{k=1}^{K}(\boldsymbol{\alpha})_k)}
\end{aligned} \tag{4.19}$$

Note that the sampling distribution for $(\mathbf{Z_d})_i$ depends only on the fixed parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, and the counts $(\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_{(\mathbf{w_d})_i}$, $(\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_j$ and $(\mathbf{m_d})_{(\mathbf{z_d})_i}$. To clarify, $(\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_{(\mathbf{w_d})_i}$ represents the number of times word $(\mathbf{w_d})_i$ is assigned to topic $(\mathbf{z_d})_i$, $(\mathbf{n}_{(\mathbf{z_d})_\mathbf{i}})_j$ equals the number of times word j is assigned to topic $(\mathbf{z_d})_i$. Lastly, $(\mathbf{m_d})_{(\mathbf{z_d})_i}$ is the number of times a word in document d is assigned to topic $(\mathbf{z_d})_i$.

The result in 4.19 shows that the conditional distribution of $(\widetilde{\mathbf{Z}}_\mathbf{d})_i$ is a Multinomial:

$$(\widetilde{\mathbf{z}}_{\mathbf{d,i}}|\mathbf{w},\mathbf{z}_{-(d,i)},\boldsymbol{\alpha},\boldsymbol{\beta}) \sim \text{Multinomial}(1,\mathbf{Y}_{d,i}) \tag{4.20}$$

Where we define $\mathbf{Y}_{d,i}$:

$$\mathbf{Y}_{d,i} = \left(\frac{(\mathbf{n_1})_{(\mathbf{w_d})_i}^{-(d,i)}+(\boldsymbol{\beta})_{(\mathbf{w_d})_i}}{\sum_{j=1}^{V}((\mathbf{n_1})_j^{-(d,i)}+(\boldsymbol{\beta})_j}\cdot\frac{(\mathbf{m_d})_1^{-(d,i)}+(\boldsymbol{\alpha})_1}{\sum_{k=1}^{K}((\mathbf{m_d})_k^{-(d,i)}+(\boldsymbol{\alpha})_k)},\ldots,\frac{(\mathbf{n_K})_{(\mathbf{w_d})_i}^{-(d,i)}+(\boldsymbol{\beta})_{(\mathbf{w_d})_i}}{\sum_{j=1}^{V}((\mathbf{n_K})_j^{-(d,i)}+(\boldsymbol{\beta})_j)}\cdot\frac{(\mathbf{m_d})_K^{-(d,i)}+(\boldsymbol{\alpha})_K}{\sum_{k=1}^{K}((\mathbf{m_d})_k^{-(d,i)}+(\boldsymbol{\alpha})_k)}\right) \tag{4.21}$$

As mentioned before, in collapsed Gibbs sampling, initially, only the topics are sampled. After these samples are obtained, the parameters that were integrated out, here $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ can be sampled conditional on the observed variables $\mathbf{w}$, the hyperparameters $\boldsymbol{\alpha},\boldsymbol{\beta}$ and the estimated sampled variables $\mathbf{z}$) From these samples, the values of $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ can be estimated using e.g. the posterior mean. From the derivation in section 4.1.2, we know that $\left(\boldsymbol{\Theta}_\mathbf{d}|\mathbf{z},\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta}\right) \sim \text{Dirichlet}(\mathbf{m_d}+\boldsymbol{\alpha})$ and $\left(\boldsymbol{\Phi}_\mathbf{t}|\mathbf{z},\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta}\right) \sim \text{Dirichlet}(\mathbf{m_t}+\boldsymbol{\beta})$. The posterior means of these distributions are used as estimators.

$$\text{For } d\in\{1,\ldots,M\} \text{ and } j\in\{1,\ldots,K\}: \qquad (\hat{\boldsymbol{\theta}}_\mathbf{d})_j = \frac{(\mathbf{m_d})_j+(\boldsymbol{\alpha})_j}{\sum_{k=1}^{K}((\mathbf{m_d})_k+(\boldsymbol{\alpha})_k)} \tag{4.22}$$

$$\text{For } k \in \{1,\dots,K\} \text{ and } i \in \{1,\dots,V\}: \qquad (\hat{\boldsymbol{\phi}}_{\mathbf{k}})_i = \frac{(\mathbf{m_k})_i + (\boldsymbol{\beta})_i}{\sum_{j=1}^{V}((\mathbf{m_k})_j + (\boldsymbol{\beta})_j} \tag{4.23}$$

To create a better overview, the complete algorithm is given below.

Algorithm 2 Collapsed Gibbs Sampling for LDA

```
Initialize $\mathbf{z}$ and compute the initial frequencies $\mathbf{n}$ and $\mathbf{m}$
Fix $N_{iter}$ for the maximum number of iterations
for $iter = 1$ to $N_{iter}$ do
    for $d = 1$ to $M$ do                                         ▷ Iterate over documents
        for $i = 1$ to $N_d$ do                                   ▷ Iterate over words in document $d$
            Draw $(\mathbf{Z_d})_i$ from $p((\mathbf{z_d})_i|\mathbf{w}, \mathbf{z}_{-(d,i)}, \boldsymbol{\alpha}, \boldsymbol{\beta})$     ▷ Draw topic for each word
            Update $(\mathbf{n}_{(\mathbf{z_d})_i})_i$ and $(\mathbf{m_d})_{(\mathbf{z_d})_i}$
        end for
    end for
end for
Compute posterior estimates of parameters $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$
```

With the estimates of $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$, we know the per document topic-distribution and the per topic word-distribution. By determining the most frequent words per topic, one can retrieve information on the topic's theme. With the vector Θ for each document, one can decide for each document about how many topics and about which topics it tells about. Also, one can determine the, on average, most frequently mentioned topics.

As a conclusion, Markov chain Monte Carlo methods are slow in terms of convergence, such that many samples are needed for good estimations. Furthermore, in the case of LDA, we use their disadvantage of getting stuck in one topic permutation and not being able to walk through the entire domain of all latent variables. With the sample mean per latent variable, we get good estimates of $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ because we are only interested in the results of one topic permutation. The question that remains is, why computing the posterior mean using Gibbs sampling that gets stuck around one posterior mode when we can also merely determine the posterior mode? Methods to calculate the latter are elaborated on in the next section and chapter 5.

4.2. Posterior mode

Apart from the posterior mean, another estimator for the desired parameters $\mathbf{\Theta_d}$ for $d = 1,\dots,M$ and $\mathbf{\Phi_k}$ for $k = 1,\dots,K$ can be used: the posterior mode. In the paper in which LDA is introduced [7], Blei et al. use the posterior mode, but not of an approximation of the actual posterior density. This method uses variational calculus and is therefore called variational inference. It will be elaborated on in the next section.
However, the posterior mode can also be determined using the actual posterior density (up to a proportionality constant). The method to compute the posterior mode using the 'analytical' posterior density is explained in chapter 5, as it is considered a new contribution to the literature.

4.2.1. General variational methods

The posterior density of the hierarchical Bayesian model LDA can be approximated using variational methods. The posterior density of all latent variables in LDA, so including the topic assignments **Z**. Although we are only interested in Θ and Φ, the topic assignments are included in this inference method to be consistent with the application of variational methods to LDA in literature and in software.

$$
\begin{aligned}
p(\boldsymbol{\theta},\boldsymbol{\phi},\mathbf{z}|\mathbf{w}) &= \frac{p(\mathbf{w}|\boldsymbol{\theta},\boldsymbol{\phi},\mathbf{z})\,p(\boldsymbol{\theta},\boldsymbol{\phi},\mathbf{z})}{p(\mathbf{w})} \\
&= \frac{\prod_{d=1}^{M}\prod_{i=1}^{N_d}\left(p(\mathbf{\tilde{w}_{d,i}}|\boldsymbol{\theta},\boldsymbol{\phi},\mathbf{z})\right)p(\boldsymbol{\theta},\boldsymbol{\phi},\mathbf{z})}{p(\mathbf{w})} \\
&= \frac{\prod_{d=1}^{M}p(\boldsymbol{\theta}_\mathbf{d})\prod_{i=1}^{N_d}\left(\prod_{j=1}^{V}(\boldsymbol{\phi}_{(\mathbf{z_d})_i})_j^{(\mathbf{\tilde{w}_{d,i}})_j}\cdot p(\mathbf{\tilde{z}_{d,i}}|\boldsymbol{\theta}_\mathbf{d})\right)\left(\prod_{k=1}^{K}p(\boldsymbol{\phi}_\mathbf{k})\right)}{p(\mathbf{w})}
\end{aligned}
\tag{4.24}
$$

As mentioned before, the problem in computing the posterior density is the denominator in which there is an computationally intractable integral. This argument is often given for the application of variational methods. Besides, the numerator has a complicated form that cannot be traced back to a simple multivariate distribution. One can observe the coupling of $\mathbf{\Phi}$ and $(\mathbf{Z_d})_i$ in the product. This will cause problems when calculating the posterior mode. Note that in this posterior density, the topic assignments are still included, although they are not of main interest. In chapter 5, in which the posterior mode is calculated using the true posterior density, these latent topic assignments are integrated out in order to make posterior mode determination possible.

One way to deal with the difficult form of the posterior density is to approximate it by a function that makes statistical inference easier. The so-called variational parameters of the approximation function are chosen such that the approximation function is as close as possible to the true posterior density. When the best approximation function is found, the posterior mode of the approximation function can be determined. This posterior mode is then considered to be a good estimator for the model parameters $\mathbf{\Theta}$, $\mathbf{\Phi}$ and **Z**.

Before diving into the application of variational methods to LDA, the general mechanism is explained using an example. Consider n independent observations summarized in $\mathbf{y} = (y_1,\dots,y_n)$ with one-to-one corresponding hidden variables $\mathbf{X} = (X_1,\dots,X_n)$ that depend on parameter θ. The scheme is given in figure 4.3.

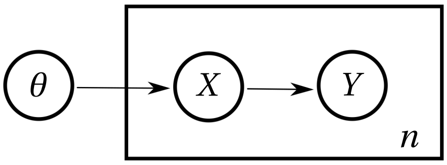


Figure 4.3: Schematic overview of an example of variational methods. There are n instances of X and Y, where X depends on the parameter θ. X is a latent random variable, θ is a fixed parameter and y is an observed random variable.

The probability of observing **y** given the parameter θ is given by:

$$
p(\mathbf{y}|\theta) = \prod_{i=1}^{n} p(y_i|\theta) = \prod_{i=1}^{n}\int_{\Omega} p(x_i, y_i|\theta)\,d\underset{i}{x}
\tag{4.25}
$$

With Ω being the set with possible outcomes of X_i. The goal in this example is to estimate parameter θ via the conditional density of $p(x_i|y_i,\theta)$ for all $i = 1,\dots,n$. This conditional density is needed, because it forms the link between data $\mathbf{y}$ and model parameter θ.

Beal notes in [1] that for models with many hidden variables, the integral in equation 4.25 can become intractable, making it difficult to compute the likelihood on the left hand side. Therefore, an approximation is determined for it, or for the log likelihood, strictly speaking. To this end, an auxiliary distribution $q_{x_i}(x_i)$ over each hidden variable x_i is introduced, where $q_{x_i}(x_i)$ can take on any form. This auxiliary distribution is an approximation for the conditional distribution $p(x_i|y_i,\theta)$:

$$\forall i \in \{1,\dots,n\}: \; q_{x_i}(x_i) \approx p(x_i|y_i,\theta) \tag{4.26}$$

Note that $q_{x_i}(x_i)$ is not a function of the observations $\mathbf{y}$. It is only an approximate density of the latent variable X_i. However, the function as a whole does depend on the observations. Namely, the auxiliary function q_{x_i} approximates the density of latent variable X_i conditional on the observation y_i and the parameter θ, i.e. $p(x_i|y_i,\theta)$. This conditional distribution is dependent on y_i. Different observations $\tilde{y}_i$ result in a different conditional distribution $p(x_i|\tilde{y}_i,\theta)$. Because q_{x_i} is an approximation of this conditional distribution, one can notice that it indeed depends on y_i, but implicitly.

The introduction of $q_{x_i}(x_i)$ can be used to derive a lower bound for the log likelihood. Note that the likelihood in equation 4.25 was intractable, so the auxiliary distributions $q_{x_i}(x_i)$ are chosen such that the lower bound for the likelihood is in fact tractable. For the sake of simplicity, the set over which x_i is integrated, Ω, is omitted.

$$\begin{aligned}
\mathscr{L}(\theta;\mathbf{y}) = \log p(\mathbf{y}|\theta) &= \log\left(\prod_{i=1}^{n}\int p(x_i,y_i|\theta)\,d\underset{i}{x}\right)\\
&= \sum_{i=1}^{n}\log\left(\int p(x_i,y_i|\theta)\,d\underset{i}{x}\right)\\
&= \sum_{i=1}^{n}\log\left(\int q_{x_i}(x_i)\frac{p(x_i,y_i|\theta)}{q_{x_i}(x_i)}\,d\underset{i}{x}\right)\\
&\geq \sum_{i=1}^{n}\int q_{x_i}(x_i)\log\left(\frac{p(x_i,y_i|\theta)}{q_{x_i}(x_i)}\right)d\underset{i}{x} \qquad *\\
&\equiv \mathscr{F}(q_{x_1}(\cdot),\dots,q_{x_n}(\cdot),\theta;\mathbf{y})
\end{aligned} \tag{4.27}$$

Where at $*$, Jensen's inequality for concave functions ($\log(x)$) is used. $\mathscr{F}$ is a functional dependent on all auxiliary functions and parameter θ. This functional $\mathscr{F}$ forms the lower bound for the log likelihood. If the lower bound equals the log likelihood, then $\forall i \in \{1,\dots,n\}$, $q_{x_i}(x_i) = p(x_i|y_i,\theta)$ and vice versa.

The optimization of the functional $\mathscr{F}$ for $q_{x_i}(\cdot)$, $\forall i \in \{1,\dots n\}$, and for parameter vector θ results in the expectation-maximization for MAP algorithm [16]. In this algorithm iteratively functions $q_{x_i}(\cdot)$ are determined given fixed θ and fixed $q_{x_j}(\cdot)$ for $j \neq i$, after which the value of θ is chosen for which the lower bound of the likelihood, i.e. $\mathscr{F}$, is maximal. This means that the two steps below are iteratively executed until convergence.

- Find $q_{x,i}(\cdot)$ by maximizing $\mathscr{F}$ for $q_{x,i}(\cdot)$ and keeping θ fixed. (E-step)
- Optimize the lower bound with respect to θ, with all auxiliary functions from the previous step substituted. This gives $\theta^{(t+1)}$. (M-step)

EM for MAP estimation is not the focus of this research, as we are not interested in estimating a fixed parameter. Remember that in Bayesian statistics, all parameters are considered random variables, except the hyperparameters. Only for the estimation of the hyperparameters, the EM for MAP method will be suitable. Therefore, this thesis will not elaborate more on this algorithm. For more information on EM for MAP estimation, one can resort to [2].

Now, we will consider a more Bayesian example. In Bayesian statistics, both the latent variables and the parameters are considered random variables. Beforehand, prior distributions on the parameter and the latent variables are imposed, after which a posterior distribution is retrieved, based on the Bayes rule. From these posterior distributions, summarizing statistics about the parameter Θ and the latent variables $\mathbf{X}$ can be

retrieved (e.g. the mean or the mode). Besides, estimators for Θ and $\mathbf{X}$ can be determined. This is a slightly different setting. Therefore, we extend the EM for MAP algorithm to the so-called Variational Bayesian EM (VBEM).
Consider the same situation as in figure 4.3. In VBEM, also for $\boldsymbol{\Theta}$ an auxiliary distribution is needed, as it is a random variable. Therefore, a general auxiliary distribution over all latent variables is introduced: $q(x_1,\ldots,x_n,\theta)$. The prior distribution of Θ depends on some fixed hyperparameter $\boldsymbol{\alpha}$, i.e. the prior has the form $p(\theta|\alpha)$.

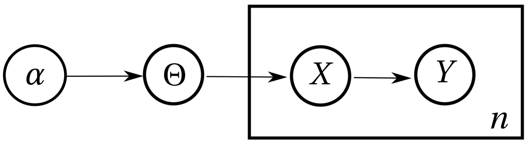


Figure 4.4: Schematic overview of an example of variational methods in a Bayesian setting. There are n instances of X and Y, where X depends on the parameter Θ. X is a latent random variable, Θ is a random variable depending on fixed hyperparameter $\boldsymbol{\alpha}$, and y is an observed random variable.

For ease of notation, we summarize the latent variables in vector $\mathbf{X} = (X_1,\ldots,X_n)$ and the observed data in $\mathbf{y} = (y_1,\ldots,y_n)$. The log likelihood of the model becomes:

$$\begin{aligned}\mathscr{L}(\alpha;\mathbf{y}) = \log p(\mathbf{y}|\alpha) &= \log\left(\int\int p(\mathbf{x},\mathbf{y},\theta|\alpha)d\mathbf{x}d\theta\right)\\ &= \log\left(\int\int q_{\mathbf{x},\theta}(\mathbf{x},\theta)\frac{p(\mathbf{x},\mathbf{y},\theta|\alpha)}{q_{\mathbf{x},\theta}(\mathbf{x},\theta)}d\mathbf{x}d\theta\right)\\ &\geq \int\int q_{\mathbf{x},\theta}(\mathbf{x},\theta)\log\left(\frac{p(\mathbf{x},\mathbf{y},\theta|\alpha)}{q_{\mathbf{x},\theta}(\mathbf{x},\theta)}\right)d\mathbf{x}d\theta \qquad *\end{aligned} \tag{4.28}$$

At $*$ again Jensen's inequality is used. With the two integral signs, it is denoted that we integrate over θ once, and over x_i for all $i = 1,\ldots,n$. Note that the difference between the likelihood $\mathscr{L}(\alpha;\mathbf{y})$ and the lower bound in equation 4.28 is exactly the Kullback-Leibler divergence of $q_{\mathbf{x},\theta}(\cdot,\cdot)$ with respect to $p(\cdot,\cdot|\mathbf{y},\alpha)$ [8].

$$\begin{aligned}\mathrm{KL}\left(q_{\mathbf{x},\theta}(\cdot,\cdot)\|p(\cdot,\cdot|\mathbf{y},\alpha)\right) &= \int\int q(\mathbf{x},\theta)\log\left(\frac{q(\mathbf{x},\theta)}{p(\mathbf{x},\theta|\mathbf{y},\alpha)}\right)d\mathbf{x}d\theta\\ &= \mathbb{E}_{q_{\mathbf{x},\theta}}\left[\log q_{\mathbf{x},\theta}(\mathbf{X},\Theta)\right] - \mathbb{E}_{q_{\mathbf{x},\theta}}\left[\log p(\mathbf{X},\Theta|\mathbf{y},\alpha)\right]\\ &= \mathbb{E}_{q_{\mathbf{x},\theta}}\left[\log q_{\mathbf{x},\theta}(\mathbf{X},\Theta)\right] - \mathbb{E}_{q_{\mathbf{x},\theta}}\left[\log\frac{p(\mathbf{X},\Theta,\mathbf{y}|\alpha)}{p(\mathbf{y}|\alpha)}\right]\\ &= \mathbb{E}_{q_{\mathbf{x},\theta}}\left[\log q_{\mathbf{x},\theta}(\mathbf{X},\Theta)\right] - \mathbb{E}_{q_{\mathbf{x},\theta}}\left[\log p(\mathbf{X},\Theta,\mathbf{y}|\alpha)\right] + \log p(\mathbf{y}|\alpha)\\ &= \mathbb{E}_{q_{\mathbf{x},\theta}}\left[\log q_{\mathbf{x},\theta}(\mathbf{X},\Theta)\right] - \mathbb{E}_{q_{\mathbf{x},\theta}}\left[\log p(\mathbf{X},\Theta,\mathbf{y}|\alpha)\right] + \mathscr{L}(\alpha;\mathbf{y})\\ &= -\int\int q_{\mathbf{x},\theta}(\mathbf{x},\theta)\log\left(\frac{p(\mathbf{x},\mathbf{y},\theta|\alpha)}{q_{\mathbf{x},\theta}(\mathbf{x},\theta)}\right)d\mathbf{x}d\theta + \mathscr{L}(\alpha;\mathbf{y})\end{aligned} \tag{4.29}$$

$$\Rightarrow \mathscr{L}(\alpha;\mathbf{y}) = \int\int q_{\mathbf{x},\theta}(\mathbf{x},\theta)\log\left(\frac{p(\mathbf{x},\mathbf{y},\theta|\alpha)}{q_{\mathbf{x},\theta}(\mathbf{x},\theta)}\right)d\mathbf{x}d\theta + \mathrm{KL}\left(q_{\mathbf{x},\theta}(\cdot,\cdot)\|p(\cdot,\cdot|\mathbf{y},\alpha)\right) \tag{4.30}$$

Note that in the derivation above, the notation $\mathbb{E}_{q_{\mathbf{x},\theta}}$ is used. The subscript $q_{\mathbf{x},\theta}$ is used to indicate that the expectation of a function of random variables $\mathbf{X}$ and Θ is computed with the joint density of $\mathbf{X}$ and Θ being $q_{\mathbf{x},\theta}$. From equation 4.30, it can be concluded that minimizing the KL-divergence of $q_{\mathbf{x},\theta}(\cdot,\cdot)$ with respect to $p(\cdot,\cdot|\mathbf{y},\alpha)$ is equivalent to maximizing the lower bound given in equation 4.28. Minimization of the KL-divergence might be more intuitive when approximating one function to another. However, in the derivation of the VBEM algorithm, we will stick to maximizing the log likelihood.

There are many possible functional forms for $q_{\mathbf{x},\theta}(\cdot,\cdot)$, but the most frequently used choice is the mean field approximation [47]. This approximation originates in the field of statistical physics [35] and assumes that all variables $X_1,\ldots,X_n,\Theta$ are independent. This is also the method used by Blei et al. in their original paper of LDA. For the example in figure 4.4, the mean field method restricts the choice in auxiliary functions to those that can be factorized such that: $q_{\mathbf{x},\theta}(\mathbf{x},\theta) \approx \left(\prod_{i=1}^{n} q_{x_i}(x_i)\right) q_\theta(\theta)$. With this choice for the auxiliary distributions, the lower bound for the log likelihood given the Bayesian model becomes:

$$\begin{aligned}\mathscr{L}(\alpha;\mathbf{y}) &\geq \int\int q_{\mathbf{x}}(\mathbf{x}) q_\theta(\theta) \log\left(\frac{p(\mathbf{x},\mathbf{y},\theta|\alpha)}{q_{\mathbf{x}}(\mathbf{x}) q_\theta(\theta)}\right) d\mathbf{x} d\theta \\ &\equiv \mathscr{F}_\alpha(q_{\mathbf{x}}(\cdot), q_\theta(\cdot);\mathbf{y})\end{aligned} \tag{4.31}$$

where we denoted $q_{\mathbf{x}}(\mathbf{x}) = \prod_{i=1}^{n} q_{x_i}(x_i)$ for simplicity.

By choosing auxiliary distributions $q_{\mathbf{x}}(\cdot)$ and $q_\theta(\cdot)$ such that the functional $\mathscr{F}_\alpha$ is maximal, we can find an approximation of the actual likelihood and we hopefully have $q_{\mathbf{x},\theta}(\mathbf{x},\theta) \approx p(\mathbf{x},\theta|\mathbf{y},\alpha)$. The assumptions on the form of the auxiliary function are quite strong when using the mean-field approximation, so we cannot tell if the two functions are close to each other for all $\mathbf{x}$ and θ.

General expressions for the auxiliary functions for which the function $\mathscr{F}_\alpha$ is maximal are given in theorem 4.2 below, based on [1].

Theorem 4.2 (Variational Bayesian EM with mean field approximation)
Let $\boldsymbol{\alpha}$ be the hyperparameter on which random variable Θ depends and let $\mathbf{Y} = (Y_1,\ldots,Y_n)$ be independently distributed with corresponding hidden variables $\mathbf{X} = (X_1,\ldots,X_n)$. A lower bound on the model's log marginal likelihood is given by the functional:

$$\mathscr{F}_\alpha = \int\int q_{\mathbf{x}}(\mathbf{x}) q_\theta(\theta) \log\left(\frac{p(\mathbf{x},\mathbf{y},\theta|\alpha)}{q_{\mathbf{x}}(\mathbf{x}) q_\theta(\theta)}\right) d\mathbf{x} d\theta \tag{4.32}$$

This can be iteratively optimized by performing the following updates for the auxiliary functions with superscript (t) as iteration number:

$$\begin{aligned} q_{x_i}^{(t+1)}(x_i) &= \frac{1}{\mathcal{Z}_{x_i}} \exp\left[\int q_\theta^{(t)}(\theta) \log\left(p(x_i, y_i|\theta,\alpha)\right) d\theta\right] \\ q_\theta^{(t+1)}(\theta) &= \frac{1}{\mathcal{Z}_\theta} \cdot p(\theta|\alpha) \cdot \exp\left[\int q_{\mathbf{x}}^{(t)}(\mathbf{x}) \log\left(p(\mathbf{x},\mathbf{y}|\theta,\alpha)\right) d\mathbf{x}\right] \\ \text{where } q_{\mathbf{x}}^{(t+1)}(\mathbf{x}) &= \prod_{i=1}^{n} q_{x_i}^{(t+1)}(x_i) \end{aligned} \tag{4.33}$$

These update rules converge to a local maximum of $\mathscr{F}_\alpha(q_{\mathbf{x}}(\cdot), q_\theta(\cdot))$. Note that $\mathcal{Z}_{x_i}$ and $\mathcal{Z}_\theta$ are normalization constants, such that the auxiliary functions integrate to 1.

Proof
In this proof, we only show the derivation of the update equations for one variable x. It can be easily seen that the result can be extended to the case in which $\mathbf{x} = (x_1,\ldots,x_n)$ and thus $q_{\mathbf{x}}(\mathbf{x}) = \prod_{i=1}^{n} q_{\mathbf{x_i}}(x_i)$. Lagrange multipliers are introduced to make sure that q_x and q_θ are valid densities, thus integrating to 1. Note that λ_x and λ_θ are strictly positive.

$$\tilde{\mathscr{F}}_\alpha = \int\left(\int q_x(x) q_\theta(\theta) \log\left(\frac{p(x,\mathbf{y},\theta|\alpha)}{q_x(x) q_\theta(\theta)}\right) d\theta\right) dx - \lambda_x\left(\int q_x(x)\, dx - 1\right)^2 - \lambda_\theta\left(\int q_\theta(\theta) d\theta - 1\right)^2 \tag{4.34}$$

First, we integrate with respect to λ_x and λ_θ and equate these derivatives to zero. By construction, this results in respectively:

$$\int q_x(x)\, dx = 1 \qquad \& \qquad \int q_\theta(\theta) d\theta = 1 \tag{4.35}$$

Then, the lower bound $\tilde{\mathscr{F}}_\alpha$ with the Lagrange multiplier terms is differentiated with respect to q_x. The definition of differentiating a functional with respect to a function can be found in appendix A.1. The functional can be rewritten as:

$$\begin{aligned}\tilde{\mathscr{F}}_\alpha &= \int\left(\int q_x(x)q_\theta(\theta)\log\left(\frac{p(x,\mathbf{y},\theta|\alpha)}{q_x(x)q_\theta(\theta)}\right)d\theta\right)dx - \lambda_x\left(\int q_x(x)\,dx - 1\right)^2 - \lambda_\theta\left(\int q_\theta(\theta)d\theta - 1\right)^2 \\ &= \int\left(\int q_x(x)q_\theta(\theta)\log\left(\frac{p(x,\mathbf{y}|\theta,\alpha)}{q_x(x)}\right)d\theta\right)dx + \int\int q_x(x)q_\theta(\theta)\log\left(\frac{p(\theta|\alpha)}{q_\theta(\theta)}\right)d\theta\,dx - \lambda_x\left(\int q_x(x)\,dx - 1\right)^2 \\ &\quad - \lambda_\theta\left(\int q_\theta(\theta)d\theta - 1\right)^2 \\ &= \int L^{(1)}(q_x,\dot{q}_x,x)\,dx + \int L^{(2)}(q_x,\dot{q}_x,x)\,dx - \lambda_x\left(\int q_x(x)\,dx - 1\right)^2 - \lambda_\theta\left(\int q_\theta(\theta)d\theta - 1\right)^2\end{aligned} \tag{4.36}$$

The differential of functional $\tilde{\mathscr{F}}_\alpha$ with respect to q_x, holding $q_\theta(\theta)$ fixed, is calculated as follows.

$$\begin{aligned}\frac{\partial\tilde{\mathscr{F}}_\alpha}{\partial q_x} &= L^{(1)}_{q_x}(q_x,\dot{q}_x,x) - \frac{d}{dx}L^{(1)}_{\dot{q}_x}(q_x,\dot{q}_x,x) + L^{(2)}_{q_x}(q_x,\dot{q}_x,x) - \frac{d}{dx}L^{(2)}_{\dot{q}_x}(q_x,\dot{q}_x,x) - \lambda_x\cdot 2\left(\int q_x(x)\,dx - 1\right)\cdot 1 \\ &= \int q_\theta(\theta)\log\big(p(x,\mathbf{y}|\theta,\alpha)\big)\,d\theta - \int q_\theta(\theta)\log\big(q_x(x)\big)\,d\theta - \int q_x(x)q_\theta(\theta)\frac{1}{q_x(x)}d\theta \\ &\quad + \int q_\theta(\theta)\log\left(\frac{p(\theta|\alpha)}{q_\theta(\theta)}\right)d\theta + 2\lambda_x\cdot\left(\int q_x(x)\,dx - 1\right) \\ &= \mathbb{E}_{q_\theta}\left[\log\big(p(x,\mathbf{y}|\Theta,\alpha)\big)\right] - \log q_x(x) - 1 + \mathbb{E}_{q_\theta}\left[\log\left(\frac{p(\theta|\alpha)}{q_\theta(\theta)}\right)\right] \qquad (*) \\ &= 0 \\ \Rightarrow q_x(x) &\propto \exp\left\{\mathbb{E}_{q_\theta}\left[\log\big(p(x,\mathbf{y}|\Theta,\alpha)\big)\right]\right\}\end{aligned} \tag{4.37}$$

Here L_{q_x} represents the derivative of functional L with respect to function q_x. At $(*)$, we used the results from equation 4.35. Furthermore, with $\mathbb{E}_{q_\theta}$ we denote the expectation of random variable Θ, where Θ has density function q_θ.

As we considered a single random variable X in this derivation, it can easily be seen that indeed, in the case with multiple latent variables X_i, for $i = 1,\ldots,n$:

$$q_{x_i}(x_i) = \frac{1}{\mathcal{Z}_{x_i}}\exp\left\{\mathbb{E}_{q_\theta}\left[\log\big(p(x_i,\mathbf{y}|\Theta,\alpha)\big)\right]\right\} \tag{4.38}$$

A similar procedure can be followed to derive the auxiliary distribution $q_\theta(\theta)$ for which the functional $\tilde{\mathscr{F}}_\alpha$ is maximal. The same steps as in equation 4.36 are followed.

$$\begin{aligned}\tilde{\mathscr{F}}_\alpha &= \int\left(\int q_x(x)q_\theta(\theta)\log\left(\frac{p(x,\mathbf{y},\theta|\alpha)}{q_x(x)q_\theta(\theta)}\right)dx\right)d\theta - \lambda_x\left(\int q_x(x)\,dx - 1\right)^2 - \lambda_\theta\left(\int q_\theta(\theta)d\theta - 1\right)^2 \\ &= \int\left(\int q_x(x)q_\theta(\theta)\log\big(p(x,\mathbf{y}|\theta,\alpha)\big)\,dx + \int q_x(x)q_\theta(\theta)\log\left(\frac{p(\theta|\alpha)}{q_\theta(\theta)}\right)dx\right)d\theta - \int\int q_\theta(\theta)q_x(x)\log\big(q_x(x)\big)\,dx\,d\theta \\ &\quad - \lambda_x\left(\int q_x(x)\,dx - 1\right)^2 - \lambda_\theta\left(\int q_\theta(\theta)d\theta - 1\right)^2 \\ &= \int L^{(1)}(q_\theta,\dot{q}_\theta,\theta)d\theta + \int L^{(2)}(q_\theta,\dot{q}_\theta,\theta)d\theta + \int L^{(3)}(q_\theta,\dot{q}_\theta,\theta)d\theta - \lambda_x\left(\int q_x(x)\,dx - 1\right)^2 - \lambda_\theta\left(\int q_\theta(\theta)d\theta - 1\right)^2\end{aligned} \tag{4.39}$$

The differential of functional $\tilde{\mathscr{F}}_\alpha$ with respect to q_θ, holding $q_x(x)$ fixed, and assuming that $q_x(x)$ is a density, is calculated as follows:

$$\begin{aligned}\frac{\partial\tilde{\mathscr{F}}_\alpha}{\partial q_\theta} &= \int q_x(x)\log p(x,\mathbf{y}|\theta,\alpha)\,dx + \int q_x(x)\log\left(\frac{p(\theta|\alpha)}{q_\theta(\theta)}\right)dx + \int q_x(x)q_\theta(\theta)\frac{q_\theta(\theta)}{p(\theta|\alpha)}\cdot\frac{-p(\theta|\alpha)}{(q_\theta(\theta))^2}\,dx \\ &\quad - \int q_x(x)\log\big(q_x(x)\big)\,dx - 2\lambda_\theta\left(\int q_\theta(\theta) - 1\right) \\ &= \int q_x(x)\log p(x,\mathbf{y}|\theta,\alpha)\,dx + \log\left(\frac{p(\theta|\alpha)}{q_\theta(\theta)}\right) - 1 - \int q_x(x)\log\big(q_x(x)\big)\,dx \qquad (*) \\ &= 0 \\ \Rightarrow q_\theta(\theta) &\propto p(\theta|\alpha)\cdot\exp\left\{\mathbb{E}_{q_x}\left[\log p(X,\mathbf{y}|\theta,\alpha)\right]\right\} \qquad (**)\end{aligned} \tag{4.40}$$

At $(*)$, we used the results from 4.35, and at $(**)$ the proportionality sign arises from the fact that we are only interested in the terms that contain θ. The update equation for q_θ becomes:

$$q_\theta(\theta) = \frac{1}{\mathcal{Z}_\theta} p(\theta|\alpha) \cdot \exp\left\{\mathbb{E}_{q_\mathbf{x}}\left[\log p(\mathbf{X}, \mathbf{y}|\theta, \alpha)\right]\right\} \tag{4.41}$$

The functional derivatives are computed under the assumption that $\tilde{\mathscr{F}}_\alpha$ is smooth and differentiable. Also we assume that there is a local maximum. At least, it is known that a maximum of $\tilde{\mathscr{F}}_\alpha$ exists, as it is bounded from above by the log likelihood. The precise proof of this convergence needs more work and is left for future research. ■

In a less formally described way, variational Bayesian EM can be explained as follows. Suppose the posterior density is intractable. We introduce auxiliary distributions over the latent variables. The auxiliary distributions are chosen in such a way that the lower bound for the likelihood of the observed data is as tight as possible to the actual likelihood. This means that the product of all auxiliary densities is an approximation of the actual posterior density of all latent variables and random parameters given the data. That is, the posterior density of all variables of interest. Lastly, with these approximate distributions, the values of the latent variables and parameters easily can be estimated using, e.g. the posterior mean or the posterior mode of each auxiliary distribution.

4.2.2. Variational Bayesian EM for LDA

The variational Bayesian EM algorithm is used for inference in the paper by Blei et al. [7]. However, we will follow the derivation in [8], because in this paper variational Bayes is applied to the so-called smoothed LDA from [7], which corresponds to how LDA is defined in this thesis, that is with a prior distribution on topic-word distributions $\mathbf{\Phi}$. In theorem 4.2, variational Bayesian EM is defined using the terminology of latent variables and parameters. In LDA these can be considered the same, as the hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are fixed and do not need statistical inference. All $\mathbf{\Phi}$, $\mathbf{\Theta}$ and $\mathbf{Z}$ are latent variables.
The log likelihood of the latent variables in LDA and the lower bound are given by 4.42. For ease of notation and because they are fixed, the conditioning on the hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is omitted.

$$\begin{aligned}\mathscr{L}(\boldsymbol{\alpha}, \boldsymbol{\beta}; \mathbf{w}) = \log p(\mathbf{w}) &= \log\left(\int\int\sum_{\mathbf{z}} p(\boldsymbol{\phi}, \boldsymbol{\theta}, \mathbf{z}, \mathbf{w}) d\boldsymbol{\theta} d\boldsymbol{\phi}\right) \\ &\geq \int\int\sum_{\mathbf{z}} q(\boldsymbol{\phi}, \boldsymbol{\theta}, \mathbf{z}) \log\frac{p(\boldsymbol{\phi}, \boldsymbol{\theta}, \mathbf{z}, \mathbf{w})}{q(\boldsymbol{\phi}, \boldsymbol{\theta}, \mathbf{z})} d\boldsymbol{\theta} d\boldsymbol{\phi}\end{aligned} \tag{4.42}$$

In [8], it is assumed that $p(\boldsymbol{\phi}, \boldsymbol{\theta}, \mathbf{z}|\mathbf{w})$ can be approximated by a mean field variational family consisting of auxiliary distributions. Note that in the expressions for latent variables $\mathbf{Z}$, the topics, a simple Z is used, representing the topic index. Actually, the topic is distributed as $\tilde{\mathbf{Z}} \sim \text{Multinomial}(1, \mathbf{\Theta})$, such that the only non-zero component of vector $\tilde{\mathbf{Z}}$ gives the value of topic Z (e.g. $\tilde{\mathbf{Z}} = \mathbf{e_6} \Rightarrow Z = 6$). For simplicity, we called the vector with all topics $\mathbf{z}$, but when looking at its distribution, $\tilde{\mathbf{Z}}$ can better be used. The mean field approximation for the latent variables becomes:

$$q(\boldsymbol{\phi}, \boldsymbol{\theta}, \mathbf{z}) = \prod_{k=1}^{K} q_{\boldsymbol{\phi}_\mathbf{k}}(\boldsymbol{\phi}_\mathbf{k}; \boldsymbol{\lambda}_\mathbf{k}) \prod_{d=1}^{D} q_{\boldsymbol{\theta}_\mathbf{d}}(\boldsymbol{\theta}_\mathbf{d}; \boldsymbol{\gamma}_\mathbf{d}) \prod_{i=1}^{N_d} q_{\tilde{\mathbf{z}}_\mathbf{d,i}}(\tilde{\mathbf{z}}_\mathbf{d,i}; \boldsymbol{\nu}_\mathbf{d,i}) \tag{4.43}$$

Each auxiliary distribution is chosen to come from the same family of distributions as the one to which the conditional distribution of each latent variable conditioned on all other variables in the model belongs. Note that these conditional distributions have already been derived in section 4.1.2. This means that:

$$\begin{aligned} q_{\boldsymbol{\phi}_\mathbf{k}}(\boldsymbol{\phi}_\mathbf{k}; \boldsymbol{\lambda}_\mathbf{k}) &\leftarrow \text{Dirichlet}(\boldsymbol{\lambda}_\mathbf{k}) \\ q_{\boldsymbol{\theta}_\mathbf{d}}(\boldsymbol{\theta}_\mathbf{d}; \boldsymbol{\gamma}_\mathbf{d}) &\leftarrow \text{Dirichlet}(\boldsymbol{\gamma}_\mathbf{d}) \\ q_{\tilde{\mathbf{z}}_\mathbf{d,i}}(\tilde{\mathbf{z}}_\mathbf{d,i}; \boldsymbol{\nu}_\mathbf{d,i}) &\leftarrow \text{Multinomial}(1, \boldsymbol{\nu}_\mathbf{d,i}) \end{aligned} \tag{4.44}$$

In order to be consistent with theorem 4.2, the lower bound of $\mathscr{L}(\boldsymbol{\alpha},\boldsymbol{\beta};\mathbf{w})$ is denoted with $\mathscr{F}$.

$$\begin{aligned}
\mathscr{F}\left(q_{\boldsymbol{\phi}}(\cdot),q_{\boldsymbol{\theta}}(\cdot),q_{\mathbf{z}}(\cdot)\right) &= \int\int\sum_{\mathbf{z}} q(\boldsymbol{\phi},\boldsymbol{\theta},\mathbf{z})\log\left(\frac{p(\boldsymbol{\phi},\boldsymbol{\theta},\mathbf{z},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta})}{q(\boldsymbol{\phi},\boldsymbol{\theta},\mathbf{z})}\right)d\boldsymbol{\theta}\,d\boldsymbol{\phi}\\
&= \mathbb{E}_{q_{\phi,\theta,\mathbf{z}}}\left[\log p(\boldsymbol{\Phi},\boldsymbol{\Theta},\mathbf{Z},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta})\right]-\mathbb{E}_{q_{\phi,\theta,\mathbf{z}}}\left[\log q(\boldsymbol{\Phi},\boldsymbol{\Theta},\mathbf{Z})\right]\\
&= \mathbb{E}_{q_{\boldsymbol{\theta}}}\left[\sum_{d=1}^{M}\log p(\boldsymbol{\Theta}_{\mathbf{d}}|\boldsymbol{\alpha})\right]+\mathbb{E}_{q_{\boldsymbol{\phi}}}\left[\sum_{k=1}^{K}\log p(\boldsymbol{\Phi}_{\mathbf{k}}|\boldsymbol{\beta})\right]+\mathbb{E}_{q_{\boldsymbol{\theta},\mathbf{z}}}\left[\sum_{d=1}^{M}\sum_{i=1}^{N_d}\log p(\tilde{\mathbf{Z}}_{\mathbf{d,i}}|\boldsymbol{\Theta}_{\mathbf{d}})\right]\\
&\quad+\mathbb{E}_{q_{\boldsymbol{\phi},\mathbf{z}}}\left[\sum_{d=1}^{M}\sum_{i=1}^{N_d}\log p(\tilde{\mathbf{w}}_{\mathbf{d,i}}|\tilde{\mathbf{Z}}_{\mathbf{d,i}},\boldsymbol{\Phi}_{\mathbf{k}})\right]-\mathbb{E}_{q_{\boldsymbol{\phi}}}\left[\sum_{k=1}^{K}\log q(\boldsymbol{\Phi}_{\mathbf{k}}|\boldsymbol{\lambda}_{\mathbf{k}})\right]\\
&\quad-\mathbb{E}_{q_{\boldsymbol{\theta}}}\left[\sum_{d=1}^{M}\log q(\boldsymbol{\Theta}_{\mathbf{d}}|\boldsymbol{\gamma}_{\mathbf{d}})\right]-\mathbb{E}_{q_{\mathbf{z}}}\left[\sum_{d=1}^{M}\sum_{i=1}^{N_d}\log q(\tilde{\mathbf{Z}}_{\mathbf{d,i}}|\boldsymbol{\nu}_{\mathbf{d,i}})\right]
\end{aligned}\tag{4.45}$$

Substituting the known conditionals and the auxiliary distributions as proposed in 4.44:

$$\begin{aligned}
\mathscr{F}\left(q_{\boldsymbol{\phi}}(\cdot),q_{\boldsymbol{\theta}}(\cdot),q_{\mathbf{z}}(\cdot)\right) &= \sum_{d=1}^{M}\left(\log\left(\Gamma(\sum_{k=1}^{K}(\boldsymbol{\alpha})_k)\right)-\sum_{k=1}^{K}\log(\Gamma((\boldsymbol{\alpha})_k))+\sum_{k=1}^{K}((\boldsymbol{\alpha})_k-1)\cdot\mathbb{E}_{q_{\boldsymbol{\theta}}}\left[\log(\boldsymbol{\Theta}_{\mathbf{d}})_k\right]\right)\\
&\quad+\sum_{k=1}^{K}\left(\log\left(\Gamma(\sum_{j=1}^{V}(\boldsymbol{\beta})_j\right)-\sum_{j=1}^{V}\log\left(\Gamma((\boldsymbol{\beta})_j\right)+\sum_{j=1}^{V}((\boldsymbol{\beta})_j-1)\cdot\mathbb{E}_{q_{\boldsymbol{\phi}}}\left[\log(\boldsymbol{\Phi}_{\mathbf{k}})_j\right]\right)\\
&\quad+\sum_{d=1}^{M}\sum_{i=1}^{N_d}\mathbb{E}_{q_{\boldsymbol{\theta}}}\left[\mathbb{E}_{q_{\mathbf{z}}}\left[\log\left(p(\tilde{\mathbf{Z}}_{\mathbf{d,i}}|\boldsymbol{\Theta}_{\mathbf{d}})\right)\middle|\boldsymbol{\Theta}\right]\right]\\
&\quad+\sum_{d=1}^{M}\sum_{i=1}^{N_d}\mathbb{E}_{q_{\boldsymbol{\phi}}}\left[\mathbb{E}_{q_{\mathbf{z}}}\left[\log\left(p(\tilde{\mathbf{w}}_{\mathbf{d,i}}|\boldsymbol{\Phi},\tilde{\mathbf{Z}}_{\mathbf{d,i}})\right)\middle|\boldsymbol{\Phi}\right]\right]\\
&\quad-\sum_{d=1}^{M}\left(\log\left(\Gamma(\sum_{k=1}^{K}(\boldsymbol{\gamma}_{\mathbf{d}})_k)\right)-\sum_{k=1}^{K}\log\left(\Gamma((\boldsymbol{\gamma}_{\mathbf{d}})_k)\right)+\sum_{k=1}^{K}((\boldsymbol{\gamma}_{\mathbf{d}})_k-1)\cdot\mathbb{E}_{q_{\boldsymbol{\theta}}}\left[\log(\boldsymbol{\Theta}_{\mathbf{d}})_k\right]\right)\\
&\quad-\sum_{k=1}^{K}\left(\log\left(\Gamma(\sum_{j=1}^{V}(\boldsymbol{\lambda}_{\mathbf{k}})_j)\right)-\sum_{j=1}^{V}\log\left(\Gamma((\boldsymbol{\lambda}_{\mathbf{k}})_j)\right)+\sum_{j=1}^{V}((\boldsymbol{\lambda}_{\mathbf{k}})_j-1)\cdot\mathbb{E}_{q_{\boldsymbol{\phi}}}\left[\log(\boldsymbol{\Phi}_{\mathbf{k}})_j\right]\right)\\
&\quad-\sum_{d=1}^{M}\sum_{i=1}^{N_d}\sum_{k=1}^{K}\log\left((\boldsymbol{\nu}_{\mathbf{d,i}})_k\right)\mathbb{E}_{q_{\mathbf{z}}}\left[(\tilde{\mathbf{Z}}_{\mathbf{d,i}})_k\right]\\
&= C+\sum_{d=1}^{M}\sum_{k=1}^{K}\left((\boldsymbol{\alpha})_k-(\boldsymbol{\gamma}_{\mathbf{d}})_k+\sum_{i=1}^{N_d}(\boldsymbol{\nu}_{\mathbf{d,i}})_k\right)\cdot\mathbb{E}_{q_{\boldsymbol{\theta}}}\left[\log(\boldsymbol{\Theta}_{\mathbf{d}})_k\right]\\
&\quad+\sum_{k=1}^{K}\sum_{j=1}^{V}\left((\boldsymbol{\beta})_j-(\boldsymbol{\lambda}_{\mathbf{k}})_j+\sum_{d=1}^{M}\sum_{i=1}^{N_d}(\tilde{\mathbf{w}}_{\mathbf{d,i}})_j\cdot(\boldsymbol{\nu}_{\mathbf{d,i}})_k\right)\cdot\mathbb{E}_{q_{\boldsymbol{\phi}}}\left[\log(\boldsymbol{\Phi}_{\mathbf{k}})_j\right]\\
&\quad-\sum_{d=1}^{M}\left(\log\left(\Gamma(\sum_{k=1}^{K}(\boldsymbol{\gamma}_{\mathbf{d}})_k)\right)-\sum_{k=1}^{K}\log\left(\Gamma((\boldsymbol{\gamma}_{\mathbf{d}})_k)\right)\right)\\
&\quad-\sum_{k=1}^{K}\left(\log\left(\Gamma(\sum_{j=1}^{V}(\boldsymbol{\lambda}_{\mathbf{k}})_j)\right)-\sum_{j=1}^{V}\log\left(\Gamma((\boldsymbol{\lambda}_{\mathbf{k}})_j)\right)\right)\\
&\quad-\sum_{d=1}^{M}\sum_{i=1}^{N_d}\sum_{k=1}^{K}(\boldsymbol{\nu}_{\mathbf{d,i}})_k\log\left((\boldsymbol{\nu}_{\mathbf{d,i}})_k\right)
\end{aligned}\tag{4.46}$$

Differentiating $\mathscr{F}$ with respect to each variational parameter separately leads to the following update equations. Through the choice of the auxiliary distributions, we already know that they integrate to 1. Therefore, Lagrange multipliers are not needed in this case.

For the variational parameter vector belonging to $\tilde{\mathbf{Z}}_{\mathbf{d,i}}$, we get a proportionality:

$$\begin{aligned}
(\boldsymbol{\nu}_{\mathbf{d,i}})_k &\propto \exp\left\{\mathbb{E}_{q_{\boldsymbol{\theta}}}[\log(\boldsymbol{\Theta}_{\mathbf{d}})_k]+\sum_{j=1}^{V}(\tilde{\mathbf{w}}_{\mathbf{d,i}})_j\mathbb{E}_{q_{\boldsymbol{\phi}}}[\log(\boldsymbol{\Phi}_{\mathbf{k}})_j]\right\}\\
&= \exp\left\{\Psi\left((\boldsymbol{\gamma}_{\mathbf{d}})_k\right)-\Psi\left(\sum_{k=1}^{K}(\boldsymbol{\gamma}_{\mathbf{d}})_k\right)+\Psi\left((\boldsymbol{\lambda}_{\mathbf{k}})_{w_{d,i}}\right)-\Psi\left(\sum_{j=1}^{V}(\boldsymbol{\lambda}_{\mathbf{k}})_j\right)\right\}
\end{aligned}\tag{4.47}$$

Where $\Psi(\cdot)$ is the digamma function. The derivation of $\mathbb{E}_{q_{\boldsymbol{\theta}}}[\log(\boldsymbol{\Theta}_{\mathbf{d}})_k]$ is elaborated on in the appendix. The exact value of $(\boldsymbol{\nu}_{\mathbf{d,i}})_k$ is retrieved via normalization i.e. $\sum_{k=1}^{K}(\boldsymbol{\nu}_{\mathbf{d,i}})_k$ must equal 1, so one needs to divide every $(\boldsymbol{\nu}_{\mathbf{d,i}})_k$ by $\sum_{k=1}^{K}(\boldsymbol{\nu}_{\mathbf{d,i}})_k$.
The variational parameter vector belonging to $\boldsymbol{\Theta}_{\mathbf{d}}$ can be determined directly:

$$\boldsymbol{\gamma}_{\mathbf{d}} = \boldsymbol{\alpha} + \sum_{i=1}^{N_d} \boldsymbol{\nu}_{\mathbf{d,i}} \tag{4.48}$$

We obtain the variational parameter vector belonging to $\boldsymbol{\Phi}_{\mathbf{k}}$ via:

$$\boldsymbol{\lambda}_{\mathbf{k}} = \boldsymbol{\beta} + \sum_{d=1}^{M} \sum_{i=1}^{N_d} (\boldsymbol{\nu}_{\mathbf{d,i}})_k \tilde{\mathbf{w}}_{\mathbf{d,i}} \tag{4.49}$$

The method of variational Bayesian EM is summarized in algorithm 3 [8]. The E-step consists of computing the variational parameters $\boldsymbol{\gamma}$, $\boldsymbol{\nu}$ and $\boldsymbol{\lambda}$, after which the lower bound $\mathscr{F}$ is determined. These steps are executed alternately until a local maximum of $\mathscr{F}$ is attained (M-step). Only a local optimum can be found because the objective function is non-convex due to the mean-field approximation [47].

Algorithm 3 Variational Bayesian EM for LDA

```
Initialize $\boldsymbol{\gamma}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$
Compute the lower bound $\mathscr{F}$
Set $\epsilon$
Start with $i = 1$
while $|\mathscr{F}[i+1] - \mathscr{F}[i]| > \epsilon$ do
    for $d = 1$ to $M$ do                                ▷ Iterate over documents
        for $i = 1$ to $N_d$ do                          ▷ Iterate over words in document $d$
            Compute $\boldsymbol{\gamma}_{\mathbf{d}}$
            Compute $\boldsymbol{\nu}_{\mathbf{d,i}}$
        end for
    end for
    for $k = 1$ to $K$ do
        Compute $\boldsymbol{\lambda}_{\mathbf{k}}$
    end for
    Compute $\mathscr{F}[i+1]$
    $i = i + 1$
end while
Compute posterior means for $\boldsymbol{\Theta}$,$\boldsymbol{\Phi}$ and $\mathbf{Z}$ using the variational parameters $\boldsymbol{\gamma}$,$\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$.
```

When the algorithm has converged, values for the variational parameters are retrieved. With these parameters, we know the complete auxiliary distributions, whose product approximates the posterior density of all latent variables. The values of the latent variables can then be estimated using the posterior mode. The posterior mode is not determined using the true posterior density, but using the approximation by the auxiliary distributions. Because each latent variable has its own auxiliary density function, the posterior modes can be determined independently. For example, if we want to get the estimator of $\boldsymbol{\Theta}_{\mathbf{d}}$:

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_{\mathbf{d}} &= \max_{\boldsymbol{\theta}_{\mathbf{d}}} \left\{ p(\boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{z} | \mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \right\} \\
&\approx \max_{\boldsymbol{\theta}_{\mathbf{d}}} \left\{ \prod_{k=1}^{K} q_{\boldsymbol{\phi}_{\mathbf{k}}}(\boldsymbol{\phi}_{\mathbf{k}}; \boldsymbol{\lambda}_{\mathbf{k}}) \prod_{d'=1}^{D} q_{\boldsymbol{\theta}_{\mathbf{d'}}}(\boldsymbol{\theta}_{\mathbf{d'}}; \boldsymbol{\gamma}_{\mathbf{d'}}) \prod_{i=1}^{N_{d'}} q_{\tilde{\mathbf{z}}_{\mathbf{d',i}}}(\tilde{\mathbf{z}}_{\mathbf{d',i}}; \boldsymbol{\nu}_{\mathbf{d',i}}) \right\} \\
&= \max_{\boldsymbol{\theta}_{\mathbf{d}}} \left\{ q_{\boldsymbol{\theta}_{\mathbf{d}}}(\boldsymbol{\theta}_{\mathbf{d}}; \boldsymbol{\gamma}_{\mathbf{d}}) \right\} \\
&= \left(\frac{(\boldsymbol{\gamma}_{\mathbf{d}})_1 - 1}{\sum_{k=1}^{K} (\boldsymbol{\gamma}_{\mathbf{d}})_k - K}, \ldots, \frac{(\boldsymbol{\gamma}_{\mathbf{d}})_K - 1}{\sum_{k=1}^{K} (\boldsymbol{\gamma}_{\mathbf{d}})_k - K} \right) \qquad (*)
\end{aligned} \tag{4.50}$$

At $(*)$, we used the expression for the mode of a Dirichlet distributed random vector. Note that this expression is only valid if all $(\boldsymbol{\gamma}_{\mathbf{d}})_i$ with $i = 1,\ldots,K$ are larger than 1. Because the mean-field assumption is quite strong, it is not guaranteed that the posterior mode of the approximation function is close to the posterior mode of the posterior density. In chapter 8, we will visualize the variational approximation function and the true posterior density for a small example, such that more insight in gained in the functioning of the VBEM algorithm.

Zhang et al. made a complete overview of variational inference methods in [53], and concluded that research is needed on the theoretical aspects of variational inference such as the approximation errors that are involved when an approximation function replaces the posterior density. Presently, the writer of this thesis has not found quantitative methods to determine the accuracy of the variational approximation, and therefore only empirical results are shown in chapter 8.
Because one of the arguments against the application of variational inference to LDA is the fact that the mean-field approximation is too strong, improvements on this assumption are made and given in [53].
Furthermore, instead of using the update equations of 4.2, the lower bound of the likelihood, $\mathscr{F}_\alpha$ can be optimized using stochastic optimization. Still, the mean-field approximation is used on the auxiliary functions, but the optimization method is different. The version of variational inference is called Stochastic Variational inference (SVI). Further improvements on this method are made in Structured Stochastic Variational Inference, by [5]. In this method, also the mean-field approximation is let go off, and dependencies between the latent variables are allowed. These dependencies can be modeled in a hierarchical structure, thus called Hierarchical Variational Inference [38]. Another option to model dependencies is using copulas. The auxiliary function then takes the form:

$$q(\boldsymbol{\theta}) = \left(\prod_{d=1}^{M} q(\theta_d;\lambda_d)\right)\cdot c(Q(\theta_1),\ldots,Q(\theta_M)) \tag{4.51}$$

With $c(\ldots)$ a copula and each Q the cumulative distribution function corresponding to the auxiliary densities q. For more information on this extension of variational inference, we refer to [45].

Although promising improvements on variational inference and in particular variational methods applied to LDA are present in literature, the method still lacks theoretical evidence of accuracy. Furthermore, other inference methods for LDA, such as Markov chain Monte Carlo methods exist, and have better convergence results. Also, even though the posterior mode cannot easily be calculated analytically, optimization methods exist that are already proven to be useful in deep learning and neural networks. One of these optimization methods is elaborated on in the next chapter, 'Posterior mode estimation for LDA', in which we propose an optimization method that aims to find the posterior mode of the high-dimensional posterior density function of LDA.

5

Determination posterior mode estimates for LDA using optimization

In the previous chapter, different inference methods to obtain estimates of the parameters $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ (respectively the document-topic distributions and the topic-word distributions) for LDA are mentioned. The posterior mode can be determined in another way than proposed in the literature, namely via optimization methods. These techniques are often used in neural network and deep learning algorithms, and can be easily applied to hierarchical Bayesian models like LDA.
Posterior mode estimation is often referred to as MAP estimation, where MAP stands for Maximum A Posteriori (posterior mode). In figure 1.1 in the introduction, the method in this chapter is described as ‘analytical’, while we still estimate the values of $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$. The reason to still call this optimization method ‘analytical’, is the fact that the posterior density is not approximated via some method. We use the actual posterior density (up to a proportionality constant) and search for its maximum.

5.1. LDA’s posterior density

The posterior mode is given by the values of all parameters $\boldsymbol{\Theta}_\mathbf{d}$ and $\boldsymbol{\Phi}_\mathbf{k}$ (with $d = 1,\dots,M$ and $k = 1,\dots,K$) for which the posterior density in equation 5.1 is maximal. Note that the proportionality constant is not needed for this estimator. Because the posterior density is not convex in most cases, smart optimization techniques are needed.

$$p(\boldsymbol{\theta},\boldsymbol{\phi}|\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta}) \propto \left[\prod_{d=1}^{M}\prod_{j=1}^{V}\left(\sum_{k=1}^{K}(\boldsymbol{\phi}_\mathbf{k})_j(\boldsymbol{\theta}_\mathbf{d})_k\right)^{n_{d,j}}\right]\cdot\left[\prod_{d=1}^{M}\prod_{k=1}^{K}(\boldsymbol{\theta}_\mathbf{d})_k^{(\boldsymbol{\alpha})_k-1}\right]\cdot\left[\prod_{k=1}^{K}\prod_{j=1}^{V}(\boldsymbol{\phi}_\mathbf{k})_j^{(\boldsymbol{\beta})_j-1}\right] \tag{5.1}$$

To demonstrate the form of the posterior density and its non-convexity, we will first consider a simple case in which dimensionality is small enough for visualization.

Consider the example in which we have only one document (think of the silly document ‘*nice stupid nice*’) consisting of three words: $w_1 = 1$, $w_2 = 2$, $w_3 = 1$. There are only two possible words as the word index is either 1 or 2. This means that the vocabulary size V is 2. Furthermore, it is assumed that there are 2 possible topics, that is $K = 2$. At last, the hyperparameters are symmetric and are set to $\alpha = 0.1$ and $\beta = 1$. The parameters of the model that we want to estimate are: Θ_1, Φ_1, Φ_2, as $\Theta_1 + \Theta_2 = 1$ and $(\boldsymbol{\Phi}_\mathbf{1})_1 + (\boldsymbol{\Phi}_\mathbf{1})_2 = 1$. Thus, with Φ_1 is denoted the probability of word 1 for topic 1. Φ_2 is then the probability of word 1 for topic 2.

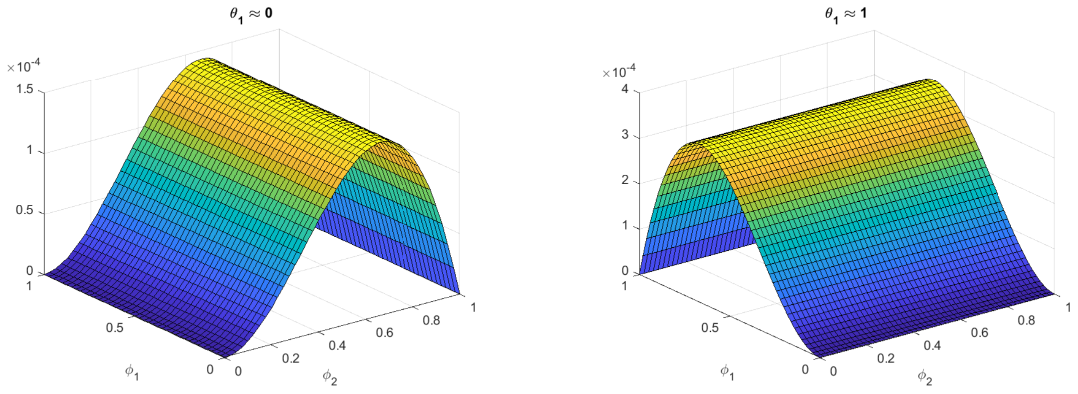
The posterior density (up to a proportionality constant) is given by:

$$\begin{aligned} p(\theta_1,\phi_1,\phi_2|\mathbf{w}) &\propto \left[\prod_{j=1}^{V}\left(\phi_{1,j}\cdot\theta_1+\phi_{2,j}\cdot(1-\theta_1)\right)^{n_{1,j}}\right]\cdot\left[\theta_1^{\alpha-1}\cdot(1-\theta_1)^{\alpha-1}\right]\left[\prod_{k=1}^{K}\phi_k^{\beta-1}\cdot(1-\phi_k)^{\beta-1}\right] \\ &= \left[\left(\phi_1\cdot\theta_1+\phi_2\cdot(1-\theta_1)\right)^2\cdot\left((1-\phi_1)\cdot\theta_1+(1-\phi_2)\cdot(1-\theta_1)\right)\right]\cdot\left[\theta_1^{-0.9}\cdot(1-\theta_1)^{-0.9}\right]\cdot 1 \end{aligned} \tag{5.2}$$

Here we take $0^0 = 1$, as is the case when $\phi_k = 0$ or $\phi_k = 1$ for some k. From equation 5.2, it can be derived that the posterior density is maximal if $\theta_1 = 0$ or $\theta_1 = 1$, as for any postive a in equation 5.3:

$$\lim_{x\downarrow 0} x^{-a} = \infty \qquad \& \qquad \lim_{x\uparrow 1}(1-x)^{-a} = \infty \tag{5.3}$$

Therefore, for $\alpha < 1$, the posterior mode will always have θ_1 drawn to the edges of the domain: $\theta_1 \in \{0,1\}$. The posterior density in equation 5.2 is shown in figure 5.1 in its two extremes for θ_1. For numerical reasons, $\theta_1 \in [\epsilon, 1-\epsilon]$, with $\epsilon = 10^{-10}$. Note that in figure 5.1, there is not one single mode, but an area in which the posterior attains the same maximum value in the form of a ridge. This is an obvious result, because in the first case where $\theta_1 \approx 0$, the document tells about topic 2, therefore nothing is known about the first topic i.e. ϕ_1. The same reasoning can be applied to figure 5.1b but the other way around. Including more documents in the analysis results, in general, in the disappearance of the ridge of posterior modes as in figure 5.1, and more distinctive modes are found.



(a) Posterior density $p(\theta_1 \approx 0, \phi_1 \phi_2 | w_1, w_2, w_3, \boldsymbol{\alpha}, \boldsymbol{\beta})$

(b) Posterior density $p(\theta_1 \approx 1, \phi_1 \phi_2 | w_1, w_2, w_3, \boldsymbol{\alpha}, \boldsymbol{\beta})$

Figure 5.1: Posterior density $p(\theta_1, \phi_1, \phi_2 | w_1, w_2, w_3, \boldsymbol{\alpha}, \boldsymbol{\beta})$ for two fixed values of θ_1 and hyperparameters $\boldsymbol{\alpha} = (0.9, 0.9)$ and $\boldsymbol{\beta} = (1,1)$.

Furthermore, it is important to note that to find the maximum, we cannot simply differentiate the posterior density to each parameter and equating the differentials to zero, as there can be saddle points. It is not easy to determine whether there are saddle points in the posterior density, but for some examples, they have been observed. Therefore, we assume that, also in high-dimensional posteriors, they are present, such that looking at the gradient being equal to 0 will not give the desired results. In the optimization method that looks for the maximum value, this will be taken into account such that it cannot get stuck in a saddle point.

Although we are now considering a simple case, already a three-dimensional matrix is involved, of size $N \times N \times N$, with N the grid size. Finding the posterior mode is the same as searching in this matrix for the maximum value(s). It is clear that for higher dimensions, the grid search for the posterior mode becomes more computationally expensive. These high dimensions are not rare, as one desires in general to find multiple topics, say 20; for accurate results many documents are taken into account, of the order 10,000; and in these documents occur many different words (even after data preprocessing), of the order 5,000. To decrease dimensionality, often the vocabulary is reduced to the 2,000 words that occur the most frequently.

Consequently, for the posterior mode we need to search into a $20 \times 10000 \times 2000 = 4 \cdot 10^8$ dimensional parameter space to find the maximal value(s) of the posterior distribution. Due to topic exchangeability, there are $K! = 20!$ posterior modes and there might be even more modes depending on the actual data. Grid search is not feasible anymore, so a smart optimization algorithm must be used. Note that optimization algorithms always search for a minimum, as is conventional. Therefore, the aim of the method is to find the minimum of the following objective.

$$\begin{aligned}\text{objective} &= -\log(\text{posterior}) \\ &= C - \log\left(\left[\prod_{d=1}^{M}\prod_{j=1}^{V}\left(\sum_{k=1}^{K}(\boldsymbol{\phi}_{\mathbf{k}})_j(\boldsymbol{\theta}_{\mathbf{d}})_k\right)^{n_{d,j}}\right]\cdot\left[\prod_{d=1}^{M}\prod_{k=1}^{K}(\boldsymbol{\theta}_{\mathbf{d}})_k^{(\boldsymbol{\alpha})_k-1}\right]\cdot\left[\prod_{k=1}^{K}\prod_{j=1}^{V}(\boldsymbol{\phi}_{\mathbf{k}})_j^{(\boldsymbol{\beta})_j-1}\right]\right) \\ &= C - \sum_{d=1}^{M}\sum_{j=1}^{V} n_{d,j}\log\left(\sum_{k=1}^{K}(\boldsymbol{\phi}_{\mathbf{k}})_j(\boldsymbol{\theta}_{\mathbf{d}})_k\right) - \sum_{d=1}^{M}\sum_{k=1}^{K}((\boldsymbol{\alpha})_k-1)\cdot\log((\boldsymbol{\theta}_{\mathbf{d}})_k) - \sum_{k=1}^{K}\sum_{j=1}^{V}\big((\boldsymbol{\beta})_j-1\big)\cdot\log\big((\boldsymbol{\phi}_{\mathbf{k}})_j\big)\end{aligned} \tag{5.4}$$

The logarithm is taken, because it makes optimization easier. Furthermore, the constant C can be omitted in the optimization, as it has no influence on the location of the minimum.

Because this thesis focuses on LDA and its statistical properties, only gradient descent optimization methods are looked into. There might be numerous other suitable optimization methods, but those are considered beyond the scope of this thesis.

5.2. Gradient descent

Gradient descent optimization is a very intuitive method of looking for a minimum. The idea is the following. Imagine you stand on a hillside, and you want to move to the lowest point in the surroundings. Because of near-sightedness and the absence of glasses, you cannot see the valley, so you need to look down and step into the direction of steepest descent. Step by step you will continue until you have reached a location in which you cannot make another step downwards. Then the local minimum is found. The idea of gradient descent is shown in figure 5.2.

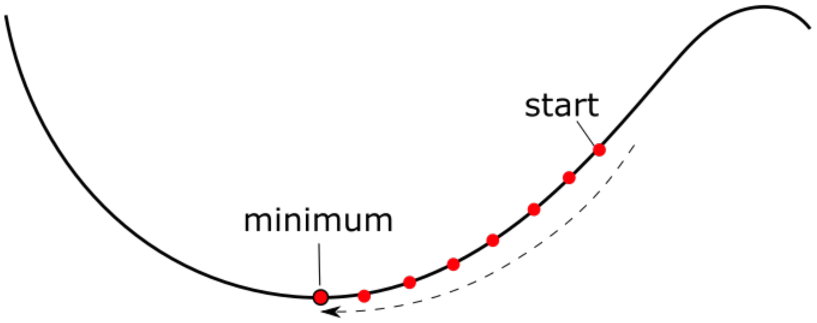


Figure 5.2: Basic gradient descent algorithm visualized. Every step is made in the direction of the steepest descent. In this one-dimensional case, one can only move upwards or downwards. Therefore, this is an easy minimization problem. The (local) minimum is reached, and no more steps can be made downwards.

Mathematically speaking, the optimization problem is the following:

$$\min_{x\in\mathbb{R}} f(x) \tag{5.5}$$

for some function $f(x)$. The gradient descent algorithm starts at an initial point $x_0 \in \mathbb{R}$ and updates via:

$$\mathbf{x_{n+1}} = \mathbf{x_n} - a\nabla f(\mathbf{x_n}) \tag{5.6}$$

until convergence ($\mathbf{x_n} \approx \mathbf{x_{n+1}}$). The step size or learning rate a needs to be chosen carefully: not too small otherwise convergence is very slow, but neither too large, to prevent jumping over the minimum. In the high-dimensional optimization problem of finding the posterior mode of equation 5.1, it is chosen to be of the order 10^{-3}.

If $f(\mathbf{x})$ is convex, the global minimum will be found. Otherwise only convergence to a local minimum is assured. [13] In high-dimensional optimization problems, computing the gradient in every iteration can be very time-consuming and computationally expensive. Therefore, one often resorts to stochastic gradient descent algorithms. [39]

5.3. Stochastic gradient descent

Stochastic gradient descent uses an update formula akin to the one used in the basic gradient descent algorithm, only a noise term is added:

$$\mathbf{x_{n+1}} = \mathbf{x_n} - a\left(\nabla f(\mathbf{x_n}) + \mathbf{w_n}\right) \tag{5.7}$$

The following proposition gives assumptions that are necessary for the algorithm to converge.

Proposition 5.1 (Stochastic gradient descent convergence, from [3])
Let $\mathbf{x_n}$ be a sequence generated by the method:

$$\mathbf{x_{n+1}} = \mathbf{x_n} + \gamma_n\left(\mathbf{s_n} + \mathbf{w_n}\right) \tag{5.8}$$

where γ_n is a deterministic positive step size, $\mathbf{s_n}$ a descent direction and $\mathbf{w_n}$ random noise. Let $\mathscr{F}_n$ be an increasing sequence of σ-fields. One can consider $\mathscr{F}_n$ to be the history of the algorithm, so it contains information about $\mathbf{x_0}, \mathbf{s_0}, \boldsymbol{\gamma_0}, \mathbf{w_0}, \ldots, \mathbf{x_{n-1}}, \mathbf{s_{n-1}}, \boldsymbol{\gamma_{n-1}}, \mathbf{w_{n-1}}$.
The function $f : \mathbb{R}^d \to \mathbb{R}$ (for some positive integer d) needs to be optimized. Furthermore, function ∇f is Lipschitz continuous with some constant L.
We assume the following:

1. *$\mathbf{x_n}$ and $\mathbf{s_n}$ are $\mathscr{F}_n$-measurable.*
2. *$\exists$ positive scalars c_1 and c_2 such that $\forall\, n$:*
$$c_1 \cdot \|\nabla f(\mathbf{x_n})\|^2 \leq -\left(\nabla f(\mathbf{x_n})\right)^T \mathbf{s_n} \qquad \& \qquad \|\mathbf{s_n}\| \leq c_2\left(1 + \|\nabla f(\mathbf{x_n})\|\right) \tag{5.9}$$
3. *For all n and with probability 1:*
$$\mathbb{E}[\mathbf{w_n}|\mathscr{F}_n] = \mathbf{0} \tag{5.10}$$
and
$$\mathbb{E}[\|\mathbf{w_n}\|^2|\mathscr{F}_n] \leq A\left(1 + \|\nabla f(\mathbf{x_n})\|^2\right) \tag{5.11}$$
where A is a positive deterministic constant.
4. *We have:*
$$\sum_{t=0}^{\infty} \gamma_n = \infty \qquad \& \qquad \sum_{t=0}^{\infty} \gamma_n^2 < \infty \tag{5.12}$$

Then, either $f(\mathbf{x_n}) \to -\infty$ or else $f(\mathbf{x_n})$ converges to a finite value and $\lim_{n\to\infty} \nabla f(\mathbf{x_n}) = 0$ almost surely. Furthermore, every limit point of $\mathbf{x_n}$ is a stationary point of f [1].

The proof of this proposition can be found in [3] and is quite extensive. Note that the fourth assumption ensures the algorithm to have steps γ_n large enough to find the stationary point of f, but at the same time not too large, such that continuing jumping over the minimum is prevented.
In the proposition, we see that for a Lipschitz continuous first derivative of function f, and noise with zero mean and bounded variance, we have almost sure convergence (or the minimum is $-\infty$). Note that the domain of function f is $\mathbb{R}^d$ in this proposition. One might get confused here, because for the posterior mode optimization problem for LDA, we want to find probability vectors that live in $(0,1)$. However, a smart transformation trick is used, such that the optimization domain is again $\mathbb{R}^d$. This trick is called the softmax transformation and is elaborated on in section 5.4.1.
The stochasticity in stochastic gradient descent is not further specified other than adding random noise to the gradient. However, other choices than random noise can be made to turn the gradient descent algorithm into stochastic gradient descent.
In the domain of deep learning, the objective function often exists of a sum of functions, i.e. $f(\mathbf{x}) = \sum_{i=1}^{m} f_i(\mathbf{x})$. Stochastic gradient descent is then defined as [3]:

$$\mathbf{x_{n+1}} = \mathbf{x_n} - a \cdot \nabla f_j(\mathbf{x}) \tag{5.13}$$

[1] A stationary point of function $f : \mathbb{R}^d \to \mathbb{R}$ is a coordinate in $\mathbb{R}^d$ in which the derivative ∇f is zero.

for some $j \in \{1,\dots,m\}$. This update formula can be rewritten in the form of equation 5.8 [3]:

$$\mathbf{x_{n+1}} = \mathbf{x_n} - a \cdot \left(\frac{1}{m} \sum_{i=1}^{m} \nabla f_i(\mathbf{x_n}) + \left[\nabla f_j(\mathbf{x_n}) - \frac{1}{m} \sum_{i=1}^{m} \nabla f_i(\mathbf{x_n}) \right] \right) \tag{5.14}$$

Note that $\frac{1}{m}\sum_{i=1}^{m} \nabla f_i(\mathbf{x_n}) = \frac{1}{m} \cdot \nabla f(\mathbf{x_n})$, so it indeed is a direction of descent. Furthermore, we can check assumption 3 from proposition 5.1 [3]:

$$\begin{aligned} \mathbb{E}[\mathbf{w_n}|\mathscr{F}_n] &= \mathbb{E}\left[\nabla f_j(\mathbf{x_n}) - \frac{1}{m} \sum_{i=1}^{m} \nabla f_i(\mathbf{x_n}) | \mathscr{F}_n \right] \\ &= \frac{1}{m} \sum_{i=1}^{m} \nabla f_i(\mathbf{x_n}) - \frac{1}{m} \sum_{i=1}^{m} \nabla f_i(\mathbf{x_n}) \\ &= 0 \end{aligned} \tag{5.15}$$

where we used the fact that j is chosen randomly and uniformly from the set $\{1,\dots m\}$. The second item in assumption 3 is the bound of the squared L^2-norm of w_n [3]:

$$\begin{aligned} \mathbb{E}[\|\mathbf{w_n}\|^2|\mathscr{F}_n] &= \mathbb{E}\left[\|\nabla f_j(\mathbf{x_n})\|^2|\mathscr{F}_n\right] - \mathbb{E}[\|w_n\||\mathscr{F}_n]^2 \\ &\le \mathbb{E}\left[\|\nabla f_j(\mathbf{x_n})\|^2|\mathscr{F}_n\right] \end{aligned} \tag{5.16}$$

Now assume that there exist positive constants C and D such that:

$$\|\nabla f_i(\mathbf{x})\| \le C + D \cdot \|\nabla f(\mathbf{x})\| \qquad \forall i, \mathbf{x}. \tag{5.17}$$

Then, it follows that:

$$\mathbb{E}[\|\mathbf{w_n}\|^2|\mathscr{F}_n] \le 2C + 2D \cdot \|\nabla f(\mathbf{x_n})\|^2 \tag{5.18}$$

which clearly satisfies equation 5.11 for adapted constant A. The other assumptions from proposition 5.1 are also satisfied as shown in [3]. As mentioned in [22], the stochasticity in this type of stochastic gradient descent does not come from random noise, but from the random selection of j for $f_j(\mathbf{x})$. Note that this form of stochastic gradient descent is called incremental gradient descent in [3]. Sometimes multiple 'sub-functions' are used instead of only f_j, to improve accuracy. Especially in high-dimensional problems with an objective that consists of a great summation, this is more accurate than taking only one sub-function. This type of gradient descent can also be referred to as mini-batch gradient descent.

The python package that is used for the implementation of posterior mode determination using optimization is called *Tensorflow*[2] and uses this mini-batch gradient descent method. Apart from computing the gradient of only a smaller sum of subfunctions of which the objective consists, also other adaptations to the basic gradient descent method are applied to increase performance. These adaptations together form the used method in this thesis: Adam optimization.

5.4. Adam optimization

Over the years and with the development of deep learning and neural networks, more high-end algorithms have been invented that speed up convergence and can better deal with non-convex objective functions and high-dimensional parameter spaces. Among them are Adadelta, RMSprop, Adagrad and Adam [39]. Adam optimization is used in this thesis to compute the posterior mode and is the most versatile for large-scale high-dimensional machine learning problems [22].

Adam is a type of stochastic gradient descent algorithm with adaptive learning rates. Its name stands for 'adaptive moment estimation' which already reveals that it uses the first and second moments of the gradient for this adaptation. We will state the algorithm for a one-dimensional problem, for ease of notation, and then explain what each step's necessity is.

[2] One of the main advantages of Tensorflow's optimization methods is that it computes gradients using automatic differentiation instead of numerical differentiation.

Algorithm 4 Adam optimization in one dimension

```
1: Set $\alpha, \beta_1, \beta_2, \epsilon$
2: Initialize $x = x_0$, $m_0 = 0$, $v_0 = 0$, $n = 0$.
3: while $\frac{f(x_n)-f(x_{n-j})}{f(x_n)}$ > threshold do
4:     $g_{n+1} = \nabla f(x_n)$
5:     $m_{n+1} = \beta_1 \cdot m_n + (1-\beta_1) \cdot g_n$
6:     $v_{n+1} = \beta_2 \cdot v_n + (1-\beta_2) \cdot g_n^2$
7:     $\hat{m}_{n+1} = \frac{m_{n+1}}{1-(\beta_1)^{n+1}}$
8:     $\hat{v}_{n+1} = \frac{v_{n+1}}{1-(\beta_2)^{n+1}}$
9:     $x_{n+1} = x_n - a \cdot \frac{\hat{m}_{n+1}}{\sqrt{\hat{v}_{n+1}}+\epsilon}$
10:    $n = n+1$
11: end while
12: Return posterior mode approximation: $x_n$
```

First, the hyperparameters are set. Recommended values are $a = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-10}$ [22]. Depending on the dimensionality of the objective, learning rate a can be adapted, because in a high dimensional problem, we want to take smaller steps than in a lower dimensional problem. The initial location in the parameter space from which the algorithm starts searching for a minimum is denoted with x_0. Then, the Adam algorithm starts 'walking' through the parameter space until convergence. Convergence is attained when the relative difference of the objective with the j-th previous value of the objective is smaller than a certain threshold. In the experiments in this thesis, for small dimensional problems, $j = 100$, and for large dimensional problems, $j = 1000$. The threshold is set to 10^{-4}, since this results in the best trade-off of accurate results within a reasonable amount of time.

In higher dimensional problems, the steps in algorithm 4 are applied to each dimension separately, that is, the gradient is determined for each dimension, there are m-terms and v-terms for each dimension, and with these dimension-specific algorithm steps, the coordinate of each dimension is updated according to step 9 in algorithm 4. For the two-dimensional case, we get $x_{n+1} = x_n - a \cdot \frac{\hat{m}_{x,n+1}}{\sqrt{\hat{v}_{x,n+1}}+\epsilon}$ and $y_{n+1} = y_n - a \cdot \frac{\hat{m}_{y,n+1}}{\sqrt{\hat{v}_{y,n+1}}+\epsilon}$.

The update step in Adam is formed by combining ideas from momentum gradient descent and the RMSprop algorithm. Let us look at the update formulas in algorithm 4 step by step.

In step 4 from algorithm 4, the gradient is computed. If x has a dimension larger than 1, the result g_n is a vector with the gradient computed with respect to each parameter dimension. Then, in step 5, a momentum term for the gradient is calculated. The formula in step 5 is a recurrence relation for the exponential moving average. Using the setting that $m_0 = 0$, we can rewrite it as:

$$m_{n+1} = (1-\beta_1) \cdot \sum_{i=1}^{n+1} (\beta_1)^{n+1-i} \cdot g_i \tag{5.19}$$

A 'basic' moving average takes into account gradients from a number of previous steps, all with equal weight. The exponential moving average is slightly different because the weights are decreasing for gradients further back in time. That is, the previous gradient has a larger influence on m_{n+1} than the gradient e.g. ten iterations back. Parameter β_1 is chosen in the interval $[0,1]$. The larger β_1, the larger the influence of previous steps. If β_1 is for example 0.5, the weight for the 10th previous iteration is only 0.001, while if $\beta_1 = 0.9$, that same weight is 0.35. This momentum term is used in optimization algorithms to damp out oscillations in the gradient. It is called a momentum term after the analogy of momentum used in physics, $\mathbf{p} = m \cdot \mathbf{v}$, with m the mass and v the velocity. One can think of a ball rolling down the slope of a bowl with initial speed not in the direction of the minimum. It will roll down towards the minimum, but its initial momentum results in a path that circles a little around the minimum.

The same momentum mechanism is applied to the gradient squared: g_n^2. Note that the square is element-wise, resulting in a vector of the same size as g_n. So, also for the gradient squared, we look at the previous iterations. Because β_2 is even larger than β_1 in the recommended settings, namely 0.999, iterations further in the past are taken into account. The reason for the computation of g_n^2 will be elaborated on in the explanation of step 9.

Then step 7 and 8 are bias correction terms. Both m_n and v_n are biased towards 0 in the first iterations of the algorithm because both have initial value 0. Therefore, if divided by respectively $1-\beta_1^n$ and $1-\beta_2^n$, they will

return larger values for small n, i.e. the first few iterations. After a certain number of iterations, the terms β_1^n and β_2^n will become that small that the bias correction step does not have any significant influence on m_n and v_n, as they are just divided by 1.
At last, the update in the parameter space is given in step 9. From the previous location x_n a step is made in the direction of the steepest descent, corrected with an exponential moving average and a bias-correction, i.e., $\hat{m}_{n+1}$. Subsequently, it is divided by the square root of the bias-corrected and exponentially averaged gradient squared term $\hat{v}_{n+1}$ plus a small constant ϵ, that is only included to avoid division by 0. This correction by $\hat{v}_{n+1}$ originates from the RMSprop algorithm [39] and results in automatic annealing, i.e., an adaptation of the learning rate.
Consider a two-dimensional parameter space. From a certain location (x_n, y_n), the gradient in the y-direction is relatively large (steep hill), while in the x-direction it is small. Then, ideally, we would like to make a large step in the x-direction, because the gradient is small and taking a large step lets us converge faster towards the minimum. On the other hand, in the y-direction, we want to take a small step, to avoid overshooting. See figure 5.3 for an illustration. Exactly this correction is made by division by $\sqrt{\hat{v}_{n+1}}$.

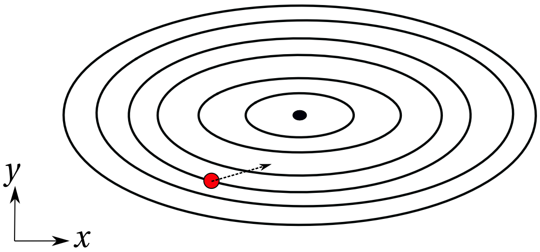


Figure 5.3: Example of finding the minimum of an ellipse-like hill. From the red dot, the gradient in the x-direction is smaller than in the y-direction, so the algorithm makes a larger step in the x-direction than in the y-direction.

The taken steps in Adam for finding the minimum of an ellipse-shaped function $f(x, y) = (x-1)^4 + 0.5y^4$ are visualized to provide more insight into the steps used in this algorithm. Note that the true minimum of $f(x, y)$ is located at $(1,0)$, and $f(x, y)$ is a convex function.

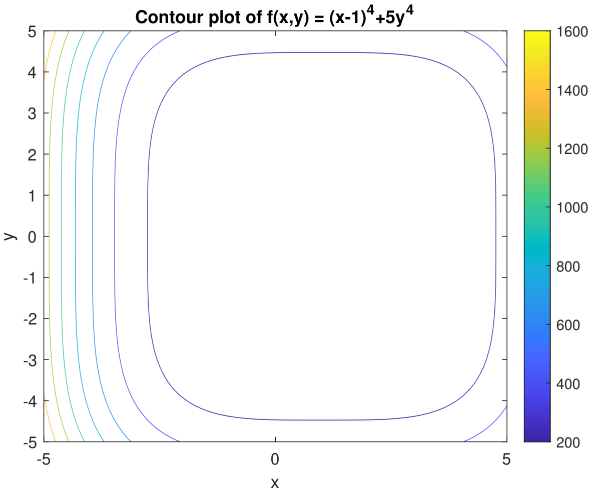


Figure 5.4: Contour plot of $f(x, y) = (x-1)^4 + 0.5y^4$. The minimum of f is located at $(x, y) = (1,0)$.

The following settings are used for the fixed parameters in Adam: $a = 0.01$, $\beta_1 = 0.8$, $\beta_2 = 0.9$, $\epsilon = 10^{-10}$. We start the search at $(x_0, y_0) = (0,2)$.

In figure 5.5, we see that the algorithm walks smoothly to $x = 1$ and $y = 0$, the location of the minimum. Initially, m_x and m_y are zero. Then, m_x decreases very fast, while m_y increases in the first few iterations. Both make sense, as we need to walk to the right (on the x-axis) for x and to the left for y from the starting point $(0,2)$. Furthermore, m_y is larger than m_x in the absolute sense. Because an exponential moving average is used for the step size via m, the steps taken by y will be relatively large, as previous (large) gradients are taken into

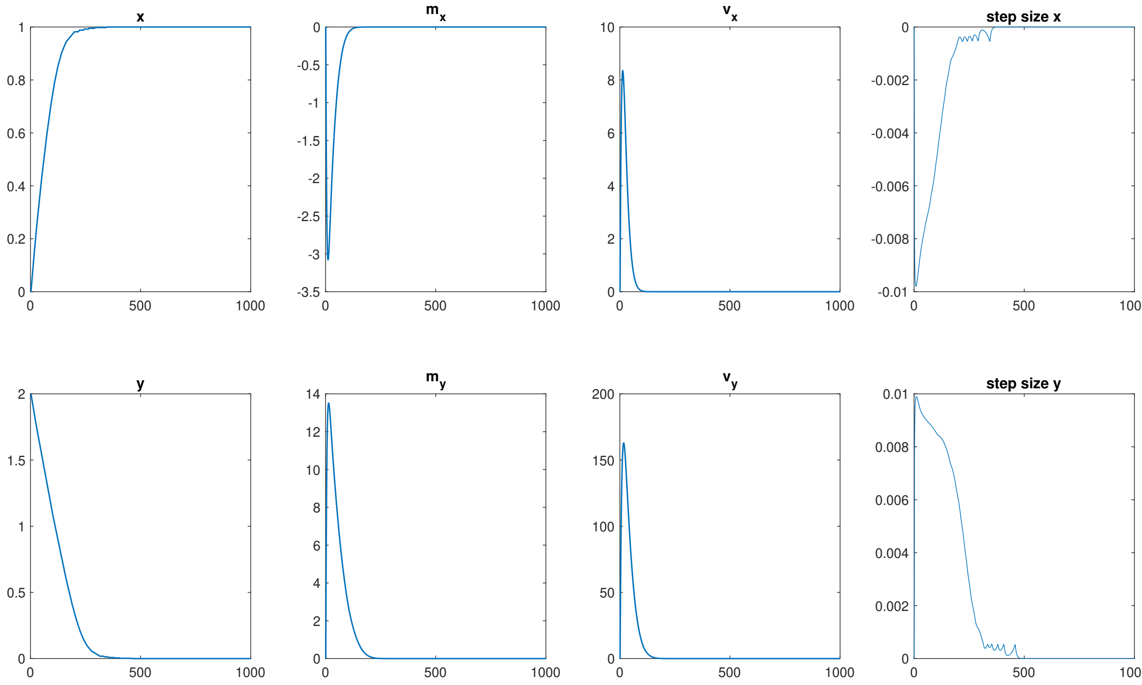


Figure 5.5: Parameters used in Adam optimization to find the location of the minimum of $f(x, y) = (x-1)^4 + 0.5y^4$. With step size is meant the size of the change in each iteration, that is: $x_{n+1} - x_n = -[\text{step size}]$ for each iteration n. If the step size is negative, the algorithm walks forwards.

account. On the other hand, we wanted to damp this effect by dividing by v, as in regions with a large gradient in the y-direction and a smaller gradient in the x-direction, the step in the former direction is smaller than the step in the latter direction. However, this phenomenon cannot clearly be seen in figure 5.5, where the step sizes of y remain large (in the absolute sense) during more iterations than the step sizes of x. Naturally, from starting point $(0, 2)$, we are further away from the minimum in the y-direction than in the x-direction. Therefore, larger steps need to be made in the y-direction, and indeed, more iterations are needed to attain the minimum in the y-direction than in the x-direction.
When we are near the optimum, the averaged gradient, m, and the averaged gradient squared, v, become very small, such that the step sizes in both x and y directions get close to zero. In the last 400 iterations, we even see that the x and y coordinates hardly change, so the minimum is attained.

In [22], the Adam algorithm is compared with three other common machine learning optimization algorithms: Adagrad, RMSprop and stochastic gradient descent with Nesterov correction. From different experiments, it can be concluded that Adam converges well, is robust and is well-suited for non-convex optimization problems [22]. Therefore, it is chosen as an appropriate optimization method to compute the posterior mode for LDA.

5.4.1. Softmax transformation

For the application of Adam optimization to LDA inference, a variable transformation is needed. The parameter space of the posterior density for LDA is $(0, 1)^P$ with P the number of parameters to be estimated. For hyperparameters $\boldsymbol{\alpha}$ and/or $\boldsymbol{\beta}$ smaller than 1, parameters reaching 0 or 1 will cause numerical problems, as the posterior goes to infinity. Furthermore, all parameter vectors $\boldsymbol{\theta}_\mathbf{d}$ and $\boldsymbol{\phi}_\mathbf{k}$ with $d = 1, \ldots, M$ and $k = 1, \ldots, K$ need to sum to 1. These two constraints are relatively hard to implement in the algorithm. Therefore, the softmax transformation is applied to all parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. This transformation is defined for, e.g. $\boldsymbol{\theta}_\mathbf{d}$ of length K, as:

$$(\boldsymbol{\theta}_\mathbf{d})_i = \frac{e^{(\tilde{\boldsymbol{\theta}}_d)_i}}{\sum_{j=1}^{K} e^{(\tilde{\boldsymbol{\theta}}_d)_j}} \tag{5.20}$$

where $(\boldsymbol{\theta}_\mathbf{d})_i$ is between 0 and 1, as desired, and $\sum_{i=1}^{K}(\boldsymbol{\theta}_\mathbf{d})_i = 1$. Now, the optimization takes place in the parameter space of $(\tilde{\boldsymbol{\theta}}_d)_i$, which is actually $\mathbb{R}$. With this transformation, both constraints are automatically implemented, and there are no numerical problems around 0 or 1.
However we need to be careful. If the posterior mode is attained for $\theta_i = 0$, as can be the case for $\boldsymbol{\alpha}$ or $\boldsymbol{\beta}$ smaller than 1, $(\tilde{\boldsymbol{\theta}}_d)_i$ must go to $-\infty$. Therefore, the optimization will keep running, pushing $\tilde{\theta}_i$ towards $-\infty$. A regularization term creates a bound on the size of $(\tilde{\boldsymbol{\theta}}_d)_i$, such that the optimization algorithm will be punished if it keeps pushing a transformation parameter, e.g. $(\tilde{\boldsymbol{\theta}}_d)_i$ towards $-\infty$. For LDA inference, we do not need parameters that are very accurate. That is, if $\theta_i = 0$ in the true posterior mode, then $\theta_i = 10^{-4}$ is more than accurate enough, especially because the number of topics K (and thus the size of $\boldsymbol{\theta}$) is rarely larger than 50.

With the softmax transformation, the solution found by Adam is not unique anymore in terms of transformed variables $\tilde{\boldsymbol{\theta}}_\mathbf{d}$ for $d = 1,\dots,M$ and $\tilde{\boldsymbol{\phi}}_\mathbf{k}$ for $k = 1,\dots,K$, that each live in $\mathbb{R}^d$ with $d = K$ for the document-topic distributions, and $d = V$ for the topic-word distributions. To each $(\tilde{\boldsymbol{\theta}}_d)_i$ a constant can be added and solution $(\boldsymbol{\theta}_d)_i$ will be the same. Exactly because the parameters we are interested in, $(\boldsymbol{\theta}_d)_i$, are invariant under these non-uniqueness of $(\tilde{\boldsymbol{\theta}}_d)_i$, we do not mind the multiple solutions. Furthermore, with regularization can be steered towards small values of $(\tilde{\boldsymbol{\theta}}_d)_i$, as will be explained in the next section.

5.4.2. Regularization

Adam optimization with the softmax transformation works well without regularization if both hyperparameters are larger than 1, as will be seen in chapter 8. However, if one of them is smaller than 1, the algorithm will keep walking towards the edges of the domain, which is $\mathbb{R}$ for each parameter if we use the softmax transformation. Regularization is applied to prevent this undesired behavior of the algorithm.
Different choices can be made for regularization. The two most common ones, especially in machine learning, are lasso and ridge regularization. The lasso ensures that the L^1-norm of each parameter vector is not too large, while in ridge regularization, the L^2-norm is used. Note that sometimes ridge regularization is referred to as Tikhonov regularization [14]. In terms of the objective, we get:

$$\text{objective} = -\log(\text{posterior}) + \lambda_x \cdot \|\mathbf{x}\|_p \tag{5.21}$$

Where for the lasso, $p = 1$, and for ridge regularization, $p = 2$. An appropriate value for λ needs to be determined iteratively. For high-dimensional problems with both α and β (much) smaller than 1, λ is chosen to be a bit larger than for lower-dimensional problems or if only one hyperparameter is smaller than 1. One can think of $\lambda = 10$ for the first case, and $\lambda = 1$ for the latter. Each parameter can have its own corresponding λ, but in practice, they are chosen to be all equal.

Both lasso and ridge regularization are implemented with Adam optimization and seem to do their jobs. However, in the extreme cases in which dimensionality is high or α or β are (much) smaller than 1, the regularization is not strong enough to prevent the algorithm from going to $-\infty$. Therefore, another stronger type of regularization is used, based on the maximum value of each random vector that is estimated. Following the example in equation 5.20 with vector $\boldsymbol{\theta}_\mathbf{d}$ for some $d \in \{1,\dots,M\}$:

$$\text{objective} = -\log(\text{posterior}) + \lambda_{\theta_d} \cdot \left(\max_i \tilde{\theta}_i\right)^4 \tag{5.22}$$

In practice, each λ is set to 1, as the regularization term itself is already strong enough to compete with the $-\log(\text{posterior})$ term in the optimization algorithm. The maximal value of each random vector is drawn to 0, as we are minimizing the objective and the maximum to the fourth power cannot be negative. Note that for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ smaller than 1, all other elements will be negative. With the maximum of each vector constrained, the other elements of that same vector will also automatically be constrained, since they are highly dependent on each other. Remember that after the softmax transformation, each random vector that is estimated sums to 1. With one element being close to 0 in the transformed space ($\mathbb{R}$), the others have to follow to prevent getting only unit vectors as estimations for $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$, which is probably not the location of the minimum of $-\log(\text{posterior})$.

6

LDA with syntax and sentiment

"The integers of language are sentences, and their organs are the parts-of-speech. Linguistic organization, then, consists in the differentiation of the parts-of-speech and the integration of the sentence."

John Wesley Powell (1834-1902)

The aim of the application of LDA to a large set of reviews is to extract about which topics people write. To this end, a topic distribution per document and a word distribution per topic are determined. The first shows in which proportions topics are written about, and therefore, which are the most important to each customer and on average. The second distribution tells what story or theme is linked to each topic. Based on the word distribution, we hope to draw a founded conclusion on the overarching issue of that topic.

Although the basic version of LDA can already be very informative, it would be better to know a sentiment per topic from the reviews. Think about knowing which words are used to describe a favorable opinion on, e.g., the price of a product. Secondly, the topic-word distributions consist of all types of words in terms of parts-of-speech: nouns, adjectives, verbs, pronouns et cetera. It would be ideal if each topic tells about one aspect of a product, linked with opinion words such that a story is told in that topic. An aspect is typically described using nouns or verbs, while opinion words are from the lexical categories adjectives or adverbs. Combining the wishes above results in the extension defined in this chapter. The results from 'LDA with syntax and sentiment' consist of topic-distributions per document, sentiment-distributions per document and, most importantly, a word-distribution per topic, sentiment, and part-of-speech combined. This means that there are $K \cdot \Sigma \cdot C$ word-distributions, where K is the number of topics, Σ is the number of possible sentiments, and C the number of different parts-of-speech used.

An extra feature of the extension from this chapter is the integration of sentence LDA. Instead of only looking at documents as with a bag of words, we focus the attention to sentences or even clauses, hereafter referred to as phrase, although, strictly speaking, this is not the best terminology. The critical and strong assumption is made that each phrase is about one topic and has one sentiment. It is expected that with this assumption, and with the construction of bags of words within each phrase instead of on document-level, the results of LDA will become more accurate.

In this chapter, first the generative model will be explained, and the plate notation of LDA with syntax and sentiment will be elaborated on. Then some small notes on practicalities are made, e.g., how to split documents into phrases? Lastly, the posterior distribution of the desired random variables is derived, and inference methods are explained.

6.1. Into the more complicated mind of the writer: generative process

In this section, we will dive again into the mind of the writer of a review, but now we assume that there are more steps involved than those described in section 3.1. We will take the example of a stroller review again.

As a writer, you first think about which aspects you want to write. Suppose you feel disappointed, as the stroller

you have bought was very expensive, and you are not satisfied with the product. Then, you want to explain your disappointment: the stroller is cumbersome, too large to fit in the car, and the basket underneath is too small. As a consequence, you want to talk about four topics: value for money, weight, size, and the basket. In addition to these topics, a sentiment is added. The value for money aspect gives you a negative sentiment. The same is valid for weight, size, and basket. When you start writing the review, in each sentence or clause, one aspect is described with negative sentiment. This supports the assumption that every phrase has only one topic and one sentiment.
Secondly, once you have chosen the topic and your opinion, words need to be selected. In general, you will use nouns or verbs to describe aspects. Think of the example above in which ‘fit’ and ‘basket’ are nouns. Then, the sentiment is often described by an adjective or adverb; ‘too large’ or ‘too small’. Note that we need the word ‘too’ here to describe a negative sentiment. If the words ‘too’ and ‘small’ occur together in many phrases, they will both have a high probability in the word distributions for that topic, and therefore the negative sentiment can be extracted.

The process of writing a review can be summarized in the following steps.

1. For each **topic** $k \in \{1,\dots,K\}$:
    (a) For each **sentiment** $o \in \{1,\dots,\Sigma\}$:
        i. Draw a topic-sentiment-word distribution $\mathbf{\Phi_{k,o}}$ from Dirichlet($\boldsymbol{\beta}_\mathbf{o}$)
2. For each **document** $d \in \{1,\dots,M\}$:
    (a) Draw a topic distribution $\mathbf{\Theta_d}$ from Dirichlet($\boldsymbol{\alpha}$)
    (b) Draw a sentiment distribution $\mathbf{\Pi_d}$ from Dirichlet($\boldsymbol{\gamma}$)
    (c) For each **phrase** $s \in \{1,\dots,S_d\}$:
        i. Draw a topic $Z_{d,s}$ from Multinomial$(1,\mathbf{\Theta_d})$
        ii. Draw a sentiment $\Sigma_{d,s}$ from Multinomial$(1,\mathbf{\Pi_d})$
        iii. For each **word** i in sentence s:
            A. Pick a part-of-speech $c_{d,s,i}$
            B. Draw a word $(\mathbf{W_{d,s}})_i$ from Multinomial$(1,\mathbf{\Phi}_{(\mathbf{Z_d})_{\mathbf{s'}},(\boldsymbol{\sigma}_\mathbf{d})_{\mathbf{s'}},\mathbf{c_{d,s,i}}})$

Again, attention must be paid to all steps in which is drawn from a Multinomial distribution. Drawing from a Multinomial$(1,\mathbf{\Theta_d})$ results in drawing a vector instead of an integer. Therefore, remember we defined in section 3.1, $\mathbf{\tilde{Z}_{d,s}} \sim$ Multinomial$(1,\mathbf{\Theta_d})$, such that $(\mathbf{Z_d})_{s'} = k \iff \tilde{\mathbf{Z}}_{d,s} = (0,0,\dots,1,0,\dots,0)$ with only one 1 on the k-th dimension of $\mathbf{\tilde{Z}_{d,s}}$. That is $\mathbf{\tilde{Z}_{d,s}}$ is the unit vector in dimension k. When $Z_{d,s}$ is drawn from Multinomial$(1,\mathbf{\Theta_d})$, actually $\mathbf{\tilde{Z}_{d,s}}$ is drawn from Multinomial$(1,\mathbf{\Theta_d})$ and the mapping $(\mathbf{\tilde{Z}_{d,s}})_k = 1 \Rightarrow (\mathbf{Z_d})_{s'} = k$ for some $k \in \{1,\dots,K\}$ is applied.

Although in the process above, a part-of-speech $c_{d,s,i}$ is drawn, we do not include it in the final model. The reason for this choice is the fact that the part-of-speech of a word cannot be learned from data, such that the topic-sentiment-word distributions per part-of-speech would be very inaccurate. One improvement could be to set the prior distribution $\boldsymbol{\beta}_\mathbf{o,c}$ for sentiment o and part-of-speech c such that words in the vocabulary corresponding to part-of-speech c have a higher probability. However, to this end, we would still need to determine the part-of-speech of each word in the vocabulary, which is similar to first finding the topic-sentiment-word distributions with all parts-of-speech included, and then, afterwards, do the split. We conclude that the latter method is a better way to determine the word distributions per topic, per sentiment and per part-of-speech, albeit for a lower dimensionality of the parameter set that needs to be inferred.

The sentiment of the reviews can be found by taking smart prior vectors $\boldsymbol{\beta}_\mathbf{o}$. There exist lists with positive and negative words in the English language (see appendix B.6). With these lists, it can be determined which words in the vocabulary of the concerned corpus are positive, and which are negative. The remaining words are considered neutral. Then, the hyperparameter vector $\boldsymbol{\beta}_{\text{pos}}$, imposed on the positive sentiment topic-word vectors $\mathbf{\Phi_{k,\mathrm{pos}}}$, are chosen such that the positive words get a higher weight than the neutral and negative words. Note that the latter are not given probability zero, following Cromwell's rule[1].

An overview of all sets, random variables and random vectors is given in table 6.1.

Table 6.1: (Random) variables used in Latent Dirichlet Allocation with syntax and sentiment.

Symbol	Meaning	Type (and size)	Space
V	Size of vocabulary	integer	$\mathbb{N}$
K	Number of topics	integer	$\mathbb{N}$
M	Number of documents in corpus	integer	$\mathbb{N}$
Σ	Number of sentiments	integer	$\mathbb{N}$
C	Number of parts-of-speech	integer	$\mathbb{N}$
S_d	Number of phrases in document d	integer	$\mathbb{N}$
N_{S_d}	Number of words in phrase S_d	integer: 1 × 1	$\mathbb{N}$
$\boldsymbol{\alpha}$	Prior belief on document-topic distribution	vector: $1 \times K$	$\mathbb{R}^K_{>0}$
$\boldsymbol{\beta}_\mathbf{o}$	Prior belief on word distribution for a sentiment	vector: $1 \times V$	$\mathbb{R}^V_{>0}$
$\boldsymbol{\gamma}$	Prior belief on document-sentiment distribution	vector: $1 \times \Sigma$	$\mathbb{R}^\Sigma_{>0}$
$\boldsymbol{\phi}_\mathbf{k,o}$	Parameter vector of multinomial word distribution for topic k, sentiment o	vector: $1 \times V$	$\mathbb{T}_V(1)$, (simplex)
$\mathbf{\Theta_d}$	Parameter vector of multinomial topic distribution for document d	vector: $1 \times K$	$\mathbb{T}_K(1)$, (simplex)
$\mathbf{\Pi_d}$	Parameter vector of multinomial sentiment distribution for document d	vector: $1 \times \Sigma$	$\mathbb{T}_\Sigma(1)$, (simplex)
$\widetilde{\mathbf{Z}}_{\mathbf{d}s}$	Unit vector in the dimension of the chosen topic for phrase s	vector: $1 \times K$	$\{0,1\}^K$
$(\mathbf{Z_d})_{s'}$	Topic (index) for phrase s in document d	integer	$\{1,\dots,K\}$
$\widetilde{\mathbf{\Sigma}}_{\mathbf{d}s}$	Unit vector in the dimension of the chosen sentiment for phrase s	vector: $1 \times \Sigma$	$\{0,1\}^\Sigma$
$(\mathbf{\Sigma_d})_{s'}$	Sentiment (index) for phrase s in document d	integer	$\{1,\dots,\Sigma\}$
$(\mathbf{w_{d,s}})_i$	Word index i corresponding to location i in phrase s from document d	integer	$\{1,\dots,V\}$

For simplicity, several assumptions on independence are made. We assume that each sentiment distribution $\mathbf{\Pi_d}$ and each topic distribution $\mathbf{\Theta_d}$ is drawn from its prior independently of the sentiment and topic distributions of other documents, and $\mathbf{\Pi_d}$ and $\mathbf{\Theta_d}$ are independent random vectors. The latter is a strict assumption which might be violated in some reviews because topics and sentiment can be correlated. However, these assumptions are needed for tractable inference.

Furthermore, the topic for each phrase is drawn independently of the previous topics of phrases in the same document. This assumption will probably not be true in most cases, as the probability of writing about the same topic in the current phrase as in the previous phrase is different from the probability of writing about that topic in the first case. Also, the sentiment of each sentence is drawn independently from the preceding sentiments in the same document.

At the deepest level, the word level, also independence assumptions are made. Each word $(\mathbf{W_{d,s}})_i$ is drawn independently of the other words in that phrase. Also, all word distributions per topic and sentiment combination, $\mathbf{\Phi_{k,o}}$ from some topic k and sentiment o, are assumed to be independent.

The resulting plate notation (c.f. figure 3.1) of LDA with syntax and sentiment is given in figure 6.1.

[1] Oliver Cromwell (1599-1658) was an English political leader, who wrote in one of his letters to the General Assembly of Scotland: *"I beseech you, in the bowels of Christ, think it possible that you may be mistaken."* [9]. Later, this quote was used by statisticians to say that you should always leave some positive probability for unexpected things to happen, and assign a probability smaller than 1 for events that are (almost) definitely occur.

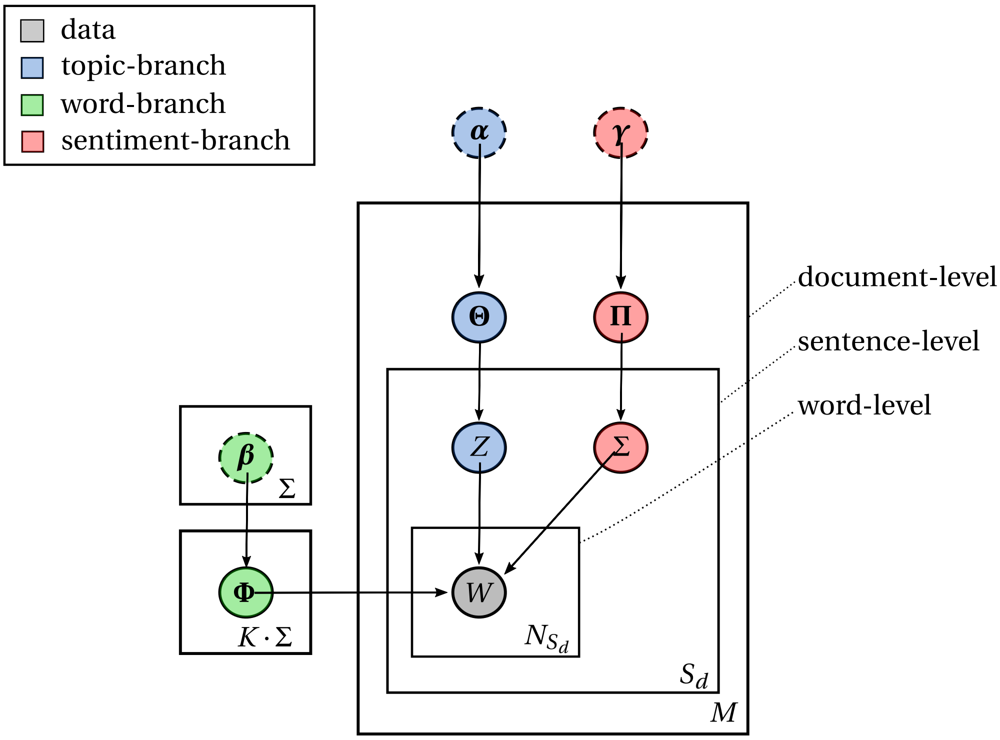


Figure 6.1: Plate notation of the extension of LDA specifically designed for review studies, also called ‘LDA with syntax and sentiment’. Each rectangle represents a repetitive action with in the right bottom corner the number of times the action (e.g. a draw from a distribution) is executed.

6.2. Practical choices in phrase detection

Because each document must be split into phrases, some rules need to be set. It is trivial that every sentence ends with a period or other kind of punctuation symbol (e.g.) ! ?). That is a natural first rule to split up a document. A comma is a more difficult punctuation mark, as it has multiple functions. Indeed, it can split up sentences into clauses, but it also arises in enumerations. In most practical implementations of this extended LDA model, the choice will be made to use every comma as a location between which phrases are split. Only attention needs to be paid, since some data sets might contain data that are not nicely written in the sense that many commas are used in each sentence. In this type of texts, it is not wise to split the documents into sentences based on comma occurrence. Therefore, a check is needed for each data set.
Lastly, conjunctions can be used to denote clauses. The conjunctions themselves are of no use in LDA, so they are only used to split up phrases and then they are removed from the data set.

An example will show how the splitting rules above function. The review below is an actual review about a shaver.

> *Nothing special with this shaver. It seems underpowered (runs on rechargeable AA batteries) and the blades aren’t high quality. It has a nice feel in your hand but it doesn’t shave nearly as close as my 10 yr. old Panasonic wet/dry. It also beat up my face a bit, leaving skin red and tender. If you have a tougher beard, I recommend investing in a higher end shaver.*

The splitting process will then be as follows. The conjunctions and punctuations are removed after the split. Also, the preprocessing steps of eliminating words that are shorter than three letters, removing numbers and all uppercase letters are converted to lowercase letters. Moreover, punctuations like the apostrophe(’) and slashes(/) are replaced with a space.

> *nothing special with this shaver | seems underpowered | runs rechargeable batteries| the blades arent high quality | has nice feel your hand | doesnt shave nearly close | old panasonic wet dry | also beat face bit | leaving skin red | tender | you have tougher beard | recommend investing higher end shaver*

The parts-of-speech that are of interest in this model are chosen to be: nouns, verbs, adverbs, and adjectives. Also, interjections are kept, although they do not occur very often. When we highlight these parts-of-speech in boldface, the aspects and corresponding sentiments in the review become clear.

> nothing **special** with this **shaver**
> **seems underpowered**
> **runs rechargeable batteries**
> the **blades arent high quality**
> **has nice feel** your **hand**
> **doesnt shave nearly close**
> **old panasonic wet dry**
> also **beat face bit**
> **leaving skin red**
> **tender**
> you **have tougher beard**
> **recommend investing higher end shaver**

From the review example above, we conclude that, theoretically, LDA with syntax and sentiment is promising and suitable for review analyses.

A remark needs to be made for this example. The review considered is written very nicely with commas where they need to be and proper English. In most review data sets, however, reviews are not written this well, and either no commas or a lot of commas are used, such that the phrases based on comma splits are not informative anymore. Therefore, one needs to decide per data set which splitting rules are the most appropriate and give the best results.

6.3. Estimating the variables of interest

LDA with syntax and sentiment aims to retrieve the topics customers write about, in combination with their sentiment about them. Also, after having retrieved the word distributions per topic and sentiment combination, a further split per part-of-speech is made, such that the final result consists of topic-sentiment-word distributions per part-of-speech, in which the following parts-of-speech are taken into account: nouns, verbs, adjectives, and adverbs.
In formulas: the goal is to retrieve estimates for $\mathbf{\Theta_d}$ and $\mathbf{\Pi_d}$ with $d = 1, \ldots, M$, and $\mathbf{\Phi_{k,o}}$ with $k = 1, \ldots, K$ and $o = 1, \ldots \Sigma$. The word distributions $\mathbf{\Phi_{k,o}}$ can then be split into word distributions per part-of-speech, i.e. $\boldsymbol{\phi}_{\mathbf{k,o,c}}$ with c being a noun, verb, adjective or adverb.

Again, different methods can be chosen to estimate the desired parameters. Because we use a Bayesian hierarchical model, it is natural to determine the posterior mean or mode. Given the topic and sentiment exchangeability, the posterior mean of the whole posterior distribution is not wise to use as an estimator for the parameters mentioned earlier. A more thorough explanation can be read in chapter 4.
Two good possibilities remain, the posterior mean via Gibbs sampling, and the posterior mode. The posterior mode estimate can be determined via Adam optimization, as described in chapter 5.

The posterior density can be expressed as follows. Note that we slightly abuse the notation $\mathbf{\Theta}$, with which we actually mean $\mathbf{\Theta_1}, \ldots, \mathbf{\Theta_M}$. The shorter notation helps to keep the posterior distribution readable.

$$\begin{aligned} p(\boldsymbol{\theta}, \boldsymbol{\pi}, \boldsymbol{\phi} | \mathbf{w}) &= \frac{p(\mathbf{w} | \boldsymbol{\theta}, \boldsymbol{\pi}, \boldsymbol{\phi}) \cdot p(\boldsymbol{\theta}, \boldsymbol{\pi}, \boldsymbol{\phi})}{p(\mathbf{w})} \\ &\propto p(\mathbf{w} | \boldsymbol{\theta}, \boldsymbol{\pi}, \boldsymbol{\phi}) \cdot p(\boldsymbol{\theta}, \boldsymbol{\pi}, \boldsymbol{\phi}) \end{aligned} \tag{6.1}$$

First, we will look at the left factor on the right-hand side, i.e. the likelihood.

$$
\begin{aligned}
p(\mathbf{w}|\boldsymbol{\theta},\boldsymbol{\pi},\boldsymbol{\phi}) &= \prod_{d=1}^{M} p(\mathbf{w_d}|\boldsymbol{\theta}_\mathbf{d},\boldsymbol{\pi}_\mathbf{d},\boldsymbol{\phi}) \\
&= \prod_{d=1}^{M}\prod_{s=1}^{S_d} p(\mathbf{w_{d,s}}|\boldsymbol{\theta}_\mathbf{d},\boldsymbol{\pi}_\mathbf{d},\boldsymbol{\phi}) \\
&= \prod_{d=1}^{M}\prod_{s=1}^{S_d}\left(\sum_{k=1}^{K} p(\mathbf{w_{d,s}}|z_{d,s}=k,\boldsymbol{\pi}_\mathbf{d},\boldsymbol{\phi}_\mathbf{k})\cdot p(z_{d,s}=k|\boldsymbol{\theta}_\mathbf{d})\right) \\
&= \prod_{d=1}^{M}\prod_{s=1}^{S_d}\left(\sum_{k=1}^{K}\sum_{o=1}^{\Sigma} p(\mathbf{w_{d,s}}|z_{d,s}=k,\ \sigma_{d,s}=o,\ \boldsymbol{\phi}_\mathbf{k,o})\cdot p(z_{d,s}=k|\boldsymbol{\theta}_\mathbf{d})\cdot p(\sigma_{d,s}=o|\boldsymbol{\pi}_\mathbf{d})\right) \\
&= \prod_{d=1}^{M}\prod_{s=1}^{S_d}\left(\sum_{k=1}^{K}\sum_{o=1}^{\Sigma}\left[\prod_{i=1}^{N_{s'}} p((\mathbf{w_{d,s}})_i|z_{d,s}=k,\ \sigma_{d,s}=o,\ \boldsymbol{\phi}_\mathbf{k,o})\right]\cdot p(z_{d,s}=k|\boldsymbol{\theta}_\mathbf{d})\cdot p(\sigma_{d,s}=o|\boldsymbol{\pi}_\mathbf{d})\right) \\
&= \prod_{d=1}^{M}\prod_{s=1}^{S_d}\left(\sum_{k=1}^{K}\sum_{o=1}^{\Sigma}\left[\prod_{i=1}^{N_{s'}}(\boldsymbol{\phi}_\mathbf{k,o})_{(\mathbf{w_{d,s}})_i}\right]\cdot(\boldsymbol{\theta}_\mathbf{d})_k\cdot(\boldsymbol{\pi}_\mathbf{d})_o\right) \\
&= \prod_{d=1}^{M}\prod_{s=1}^{S_d}\left(\sum_{k=1}^{K}(\boldsymbol{\theta}_\mathbf{d})_k\sum_{o=1}^{\Sigma}(\boldsymbol{\pi}_\mathbf{d})_o\cdot\left[\prod_{j=1}^{V}(\boldsymbol{\phi}_\mathbf{k,o})_j^{n_{d,s,j}}\right]\right)
\end{aligned}
\tag{6.2}
$$

Here, the count array n is introduced. This count array has shape $M \times \max_d\{S_d\} \times V$, such that the frequency of each word per phrase per document is registered. To obtain one array with all summarized data, we used $\max_d\{S_d\}$ as second dimension. Note that the remainder of the array is filled up with zeros, if for some document, the number of phrases is smaller than $\max_d\{S_d\}$.

Then the right term, which represents the prior distributions involved in the LDA extension can be expressed as follows.

$$
\begin{aligned}
p(\boldsymbol{\theta},\boldsymbol{\pi},\boldsymbol{\phi}) &= p(\boldsymbol{\theta}|\boldsymbol{\alpha})\cdot p(\boldsymbol{\pi}|\boldsymbol{\gamma})\cdot p(\boldsymbol{\phi}|\boldsymbol{\beta}) \\
&= \left[\prod_{d=1}^{M} p(\boldsymbol{\theta}_\mathbf{d}|\boldsymbol{\alpha})\cdot p(\boldsymbol{\pi}_\mathbf{d}|\boldsymbol{\gamma})\right]\cdot\left[\prod_{k=1}^{K}\prod_{o=1}^{\Sigma} p(\boldsymbol{\phi}_\mathbf{k,o}|\boldsymbol{\beta}_\mathbf{o})\right] \\
&\propto \left[\prod_{d=1}^{M}\left(\prod_{k=1}^{K}(\boldsymbol{\theta}_\mathbf{d})_k^{(\boldsymbol{\alpha})_k-1}\right)\left(\prod_{o=1}^{\Sigma}(\boldsymbol{\pi}_\mathbf{d})_o^{(\boldsymbol{\gamma})_o-1}\right)\right]\cdot\left[\prod_{k=1}^{K}\prod_{o=1}^{\Sigma}\prod_{j=1}^{V}(\boldsymbol{\phi}_\mathbf{k,o})_j^{(\boldsymbol{\beta}_\mathbf{o})_j-1}\right]
\end{aligned}
\tag{6.3}
$$

The complete expression for the posterior distribution of $(\boldsymbol{\Theta},\boldsymbol{\Pi},\boldsymbol{\Phi})$ then becomes:

$$
\begin{aligned}
p(\boldsymbol{\theta},\boldsymbol{\pi},\boldsymbol{\phi}|\mathbf{w}) &\propto p(\mathbf{w}|\boldsymbol{\theta},\boldsymbol{\pi},\boldsymbol{\phi})\cdot p(\boldsymbol{\theta},\boldsymbol{\pi},\boldsymbol{\phi}) \\
&= \left[\prod_{d=1}^{M}\prod_{s=1}^{S_d}\left(\sum_{k=1}^{K}(\boldsymbol{\theta}_\mathbf{d})_k\sum_{o=1}^{\Sigma}(\boldsymbol{\pi}_\mathbf{d})_o\cdot\left[\prod_{j=1}^{V}(\boldsymbol{\phi}_\mathbf{k,o})_j^{n_{d,s,j}}\right]\right)\right]\cdot\left[\prod_{d=1}^{M}\left(\prod_{k=1}^{K}(\boldsymbol{\theta}_\mathbf{d})_k^{(\boldsymbol{\alpha})_k-1}\right)\left(\prod_{o=1}^{\Sigma}(\boldsymbol{\pi}_\mathbf{d})_o^{(\boldsymbol{\gamma})_o-1}\right)\right] \\
&\quad\cdot\left[\prod_{k=1}^{K}\prod_{o=1}^{\Sigma}\prod_{j=1}^{V}(\boldsymbol{\phi}_\mathbf{k,o})_j^{(\boldsymbol{\beta}_\mathbf{o})_j-1}\right] \\
&= \left[\prod_{d=1}^{M}\left(\prod_{k=1}^{K}(\boldsymbol{\theta}_\mathbf{d})_k^{(\boldsymbol{\alpha})_k-1}\right)\left(\prod_{o=1}^{\Sigma}(\boldsymbol{\pi}_\mathbf{d})_o^{(\boldsymbol{\gamma})_o-1}\right)\left(\prod_{s=1}^{S_d}\left(\sum_{k=1}^{K}(\boldsymbol{\theta}_\mathbf{d})_k\sum_{o=1}^{\Sigma}(\boldsymbol{\pi}_\mathbf{d})_o\cdot\left[\prod_{j=1}^{V}(\boldsymbol{\phi}_\mathbf{k,o})_j^{n_{d,s,j}}\right]\right)\right)\right] \\
&\quad\cdot\left[\prod_{k=1}^{K}\prod_{o=1}^{\Sigma}\prod_{j=1}^{V}(\boldsymbol{\phi}_\mathbf{k,o})_j^{(\boldsymbol{\beta}_\mathbf{o})_j-1}\right]
\end{aligned}
\tag{6.4}
$$

6.3.1. Posterior mode: optimization

With the expression of the posterior density, we can determine the posterior mode. This statistic is expected to be a good estimator for all latent random variables of interest in the model, that is $\mathbf{\Theta_1}, \dots, \mathbf{\Theta_M}$, $\mathbf{\Pi_1}, \dots, \mathbf{\Pi_M}$ and $\mathbf{\Phi_{1,1}}, \dots, \mathbf{\Phi_{1,\Sigma}}, \dots, \mathbf{\Phi_{K,1}}, \dots, \mathbf{\Phi_{K,\Sigma}}$.

For the application of Adam optimization to find the posterior mode, the log(posterior) can better be used:

$$\begin{aligned}
\log\big(p(\boldsymbol{\theta},\boldsymbol{\pi},\boldsymbol{\phi}|\mathbf{w})\big) = C + \sum_{d=1}^{M}\Bigg[&\sum_{k=1}^{K}((\boldsymbol{\alpha})_k - 1)\log((\boldsymbol{\theta}_\mathbf{d})_k) + \sum_{o=1}^{\Sigma}((\boldsymbol{\gamma})_o - 1)\log((\boldsymbol{\pi}_\mathbf{d})_o) + \sum_{s=1}^{S_d}\log\Bigg(\sum_{k=1}^{K}(\boldsymbol{\theta}_\mathbf{d})_k\sum_{o=1}^{\Sigma}(\boldsymbol{\pi}_\mathbf{d})_o\cdot\Bigg[\prod_{j=1}^{V}(\boldsymbol{\phi}_\mathbf{k,o})_j^{n_{d,s,j}}\Bigg]\Bigg)\Bigg]\\
&+\sum_{k=1}^{K}\sum_{o=1}^{\Sigma}\sum_{j=1}^{V}((\boldsymbol{\beta}_\mathbf{o})_j - 1)\log\big((\boldsymbol{\phi}_\mathbf{k,o})_j\big)\\
= C + \sum_{d=1}^{M}\Bigg[&\sum_{k=1}^{K}((\boldsymbol{\alpha})_k - 1)\log((\boldsymbol{\theta}_\mathbf{d})_k) + \sum_{o=1}^{\Sigma}((\boldsymbol{\gamma})_o - 1)\log((\boldsymbol{\pi}_\mathbf{d})_o)\Bigg]\\
&+\sum_{d=1}^{M}\sum_{s=1}^{S_d}\log\Bigg(\sum_{k=1}^{K}(\boldsymbol{\theta}_\mathbf{d})_k\sum_{o=1}^{\Sigma}(\boldsymbol{\pi}_\mathbf{d})_o\cdot\exp\Bigg[\sum_{j=1}^{V}n_{d,s,j}\log\{(\boldsymbol{\phi}_\mathbf{k,o})_j\}\Bigg]\Bigg)\\
&+\sum_{k=1}^{K}\sum_{o=1}^{\Sigma}\sum_{j=1}^{V}((\boldsymbol{\beta}_\mathbf{o})_j - 1)\log\big((\boldsymbol{\phi}_\mathbf{k,o})_j\big)
\end{aligned} \tag{6.5}$$

Here, the constant C originates from the fact that the posterior density in equation 6.4 is only expressed up to a proportionality constant.

In equation 6.5 we can see that again the posterior density has a satisfactory form, for which optimization is well possible. The sum of subfunctions allows us to do stochastic gradient descent which is used in Adam optimization, and the python package *Tensorflow* can run the algorithm parallel, keeping computation time with reasonable bounds. Furthermore, the sum in the third term of the log(posterior) are actually three tensor products. The tensor product can be seen as a product of two high-dimensional arrays in which the dimension over which is summed is specified, and it can easily be implemented in *Tensorflow*.

The same softmax transformation and regularization methods are used as explained in chapter 5. Only an extra trick needs to be applied, as this extended version of LDA has more problems when the hyperparameters are smaller than 1.

Numerical stability via the log-sum-exp trick

The posterior density or the log posterior of LDA with syntax and sentiment is even more complicated than the posterior density in plain LDA. Therefore, other numerical problems arise. If some parameter $(\mathbf{\Theta_d})_k$, $(\mathbf{\Pi_d})_o$ or $(\mathbf{\Phi_{k,o}})_j$ gets too close to zero during the optimization process, the algorithm will reach the bounds of numerical precision when computing the exponent in the softmax transformation. The calculated objective will return a 'NaN', resulting in an immediate exit from the optimization. This problem, caused by a lack of numerical precision, can be solved by using a smart way of rewriting the log posterior.

Consider an example in which the x_i, $i = 1,\dots,n$ are such that $\log(\sum_i e^{x_i})$ is hard to compute numerically, for example due to the x_i being too negative. If e^{x_i} is smaller than the machine precision, $\log(e^{x_i})$ will return either $-\infty$ or NaN, both resulting in an objective that cannot be computed. Therefore, the optimization algorithm is stopped, and no results are given. To avoid this type of problems with machine precision, the 'log-sum-exp' transformation, often used in machine learning, is used:

$$\log\left(\sum_{i=1}^{n} e^{x_i}\right) = x^* + \log\left(\sum_{i=1}^{n} e^{x_i - x^*}\right) \tag{6.6}$$

Here, x^* is the maximum of all parameters, i.e. $x^* = \max_i\{x_i\}$. Naturally, x^* can always be calculated, as it will be of the order -50 for e^{x^*} close to 0. This is often the case when the optimization drives the optimal values of the parameters to the left bound of the domain $[0,1]$. Additionally, $e^{x_i - x^*}$ can now be computed. Where e^{-100} could not be computed, $e^{-100-(-50)}$ can, considering the example in which $x^* = -50$. Note that in the left hand side of equation 6.6, the term e^{-50} strongly dominates if all other x_j are around -100, so indeed, the right hand side is a good alternative for the computation of the log-sum-exp term.

The second term in the log posterior from equation 6.5 is rewritten with this log-sum-exp trick for each tensor product. Note that in equation 6.5, the terms are not yet of the form $\log(\sum e^{x_i})$, so some extra logarithms to facilitate the log-sum-exp trick. This might seem redundant, but it does increase numerical stability, while the optimization is still done using the same objective. In *Tensorflow* there is a ready-to-use function for this log-sum-exp trick, as it is very often used in neural network and deep learning algorithms.

6.3.2. Posterior mean: Gibbs sampling

As an alternative method of inference, Markov chain Monte Carlo sampling can be used to obtain good estimates of the model parameters. The model parameters of the LDA with syntax and sentiment model are $\boldsymbol{\Theta}_{\mathbf{d}}$ for $d = 1,\ldots,M$, the topic distribution per document, $\boldsymbol{\Pi}_{\mathbf{d}}$ also for $d = 1,\ldots,M$, the sentiment distribution per document and $\boldsymbol{\Phi}_{\mathbf{k,o}}$ for $k = 1,\ldots,K$ and $o = 1,\ldots,\Sigma$, that is, the word distribution per topic (k) and sentiment (o) combination.
In LDA with syntax and sentiment, the distributions are chosen such that Gibbs sampling is possible, that is, conditional distributions are known, and belong to a family of distributions like Dirichlet or Multinomial distributions. This makes the MCMC sampling method a lot simpler.

Although we are only interested in $\boldsymbol{\Theta}$, $\boldsymbol{\Pi}$ and $\boldsymbol{\Phi}$, all latent variables that are specified in the model need to be sampled. In LDA with syntax and sentiment, this means that $(\mathbf{Z_d})_{s'}$ and $(\boldsymbol{\Sigma}_{\mathbf{d}})_{s'}$ for $d = 1,\ldots,M$ and $s = 1,\ldots,S_d$, thus the topic and sentiment of each phrase in each document, also need to be sampled from their corresponding conditional distributions. Below, the distribution of each random variable or random vector conditional on all other random parameters in the model is derived.

First, we determine the distribution of topic distribution $\boldsymbol{\Theta}_{\mathbf{d}'}$ in document d' conditional on all other parameters in the model. The same procedure as in 4.1.2 is followed.

$$\begin{aligned}
p(\boldsymbol{\theta}_{\mathbf{d}'}|\boldsymbol{\theta}_{\mathbf{1}},\boldsymbol{\theta}_{\mathbf{d}'-\mathbf{1}},\boldsymbol{\theta}_{\mathbf{d}'+\mathbf{1}},\boldsymbol{\theta}_{\mathbf{M}},\boldsymbol{\pi},\boldsymbol{\phi},\mathbf{z},\boldsymbol{\sigma},\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}) &= \frac{p(\boldsymbol{\theta}_{\mathbf{1}},\cdots\boldsymbol{\theta}_{\mathbf{M}},\boldsymbol{\pi},\boldsymbol{\phi},\mathbf{z},\boldsymbol{\sigma},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})}{p(\boldsymbol{\theta}_{\mathbf{1}},\boldsymbol{\theta}_{\mathbf{d}'-\mathbf{1}},\boldsymbol{\theta}_{\mathbf{d}'+\mathbf{1}},\boldsymbol{\theta}_{\mathbf{M}},\boldsymbol{\pi},\boldsymbol{\phi},\mathbf{z},\boldsymbol{\sigma},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})}\\
&\propto p(\boldsymbol{\theta}_{\mathbf{1}},\cdots\boldsymbol{\theta}_{\mathbf{M}},\boldsymbol{\pi},\boldsymbol{\phi},\mathbf{z},\boldsymbol{\sigma},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})\\
&= \left(\prod_{d=1}^{M}\left[\prod_{s=1}^{S_d} p(\mathbf{w}_{\mathbf{d,s}}|\widetilde{\mathbf{z}}_{\mathbf{d,s}},\widetilde{\boldsymbol{\sigma}}_{\mathbf{d,s}},\boldsymbol{\phi}_{\mathbf{z}_{\mathbf{d,s}},\boldsymbol{\sigma}_{\mathbf{d,s}}})\cdot p(\widetilde{\mathbf{z}}_{\mathbf{d,s}}|\boldsymbol{\theta}_{\mathbf{d}})\cdot p(\widetilde{\boldsymbol{\sigma}}_{\mathbf{d,s}}|\boldsymbol{\pi}_{\mathbf{d}})\right] p(\boldsymbol{\theta}_{\mathbf{d}}|\boldsymbol{\alpha})\cdot p(\boldsymbol{\pi}_{\mathbf{d}}|\boldsymbol{\gamma})\right)\left(\prod_{k=1}^{K}\prod_{o=1}^{\Sigma} p(\boldsymbol{\phi}_{\mathbf{k,o}}|\boldsymbol{\beta}_{\mathbf{o}})\right)\\
&\propto \left(\prod_{s=1}^{S_{d'}} p(\mathbf{w}_{\mathbf{d}',\mathbf{s}}|\widetilde{\mathbf{z}}_{\mathbf{d}',\mathbf{s}},\widetilde{\boldsymbol{\sigma}}_{\mathbf{d}',\mathbf{s}},\boldsymbol{\phi}_{(\mathbf{z}_{\mathbf{d}'})_{\mathbf{s}'},(\boldsymbol{\sigma}_{\mathbf{d}'})_{\mathbf{s}'}})\,p(\widetilde{\mathbf{z}}_{\mathbf{d}',\mathbf{s}}|\boldsymbol{\theta}_{\mathbf{d}'})\right) p(\boldsymbol{\theta}_{\mathbf{d}'}|\boldsymbol{\alpha})\\
&\propto \left(\prod_{s=1}^{S_{d'}} p(\widetilde{\mathbf{z}}_{\mathbf{d}',\mathbf{s}}|\boldsymbol{\theta}_{\mathbf{d}'})\right) p(\boldsymbol{\theta}_{\mathbf{d}'}|\boldsymbol{\alpha})\\
&\propto \left(\prod_{s=1}^{S_{d'}} (\boldsymbol{\theta}_{\mathbf{d}'})_{(\mathbf{z}_{\mathbf{d}'})_{s'}}\right) p(\boldsymbol{\theta}_{\mathbf{d}'}|\boldsymbol{\alpha})\\
&= \prod_{k=1}^{K} (\boldsymbol{\theta}_{\mathbf{d}'})_k^{(\mathbf{m}_{\mathbf{d}'})_k}\cdot\prod_{k=1}^{K} (\boldsymbol{\theta}_{\mathbf{d}'})_k^{(\boldsymbol{\alpha})_k-1}\\
&= \prod_{k=1}^{K} (\boldsymbol{\theta}_{\mathbf{d}'})_k^{(\mathbf{m}_{\mathbf{d}'})_k+(\boldsymbol{\alpha})_k-1}
\end{aligned} \tag{6.7}$$

Here, $(\mathbf{m_d})_k$ is the number of times topic k is assigned to a sentence in document d. From the expression in equation 6.7, we find that:

$$\boldsymbol{\Theta}_{\mathbf{d}'}|\boldsymbol{\theta}_{\mathbf{1}},\boldsymbol{\theta}_{\mathbf{d}'-\mathbf{1}},\boldsymbol{\theta}_{\mathbf{d}'+\mathbf{1}},\boldsymbol{\theta}_{\mathbf{M}},\boldsymbol{\pi},\boldsymbol{\phi},\mathbf{z},\boldsymbol{\sigma},\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}\sim \text{Dirichlet}(\mathbf{m}_{\mathbf{d}'}+\boldsymbol{\alpha}) \tag{6.8}$$

Then, the conditional distribution of $\boldsymbol{\Pi}_{\mathbf{d}'}$, the sentiment distribution over the phrases of document d' has been derived.

$$\begin{aligned}
p(\boldsymbol{\pi}_{\mathbf{d'}}|\boldsymbol{\theta},\boldsymbol{\pi}_{\mathbf{1}},\ldots\boldsymbol{\pi}_{\mathbf{d'-1}},\boldsymbol{\pi}_{\mathbf{d'+1}},\ldots\boldsymbol{\pi}_{\mathbf{M}},\boldsymbol{\phi},\mathbf{z},\boldsymbol{\sigma},\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}) &= \frac{p(\boldsymbol{\pi}_{\mathbf{1}},\cdots\boldsymbol{\pi}_{\mathbf{M}},\boldsymbol{\theta},\boldsymbol{\phi},\mathbf{z},\boldsymbol{\sigma},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})}{p(\boldsymbol{\pi}_{\mathbf{1}},\ldots\boldsymbol{\pi}_{\mathbf{d'-1}},\boldsymbol{\pi}_{\mathbf{d'+1}},\ldots\boldsymbol{\pi}_{\mathbf{M}},\boldsymbol{\theta},\boldsymbol{\pi},\boldsymbol{\phi},\mathbf{z},\boldsymbol{\sigma},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})}\\
&\propto p(\boldsymbol{\pi}_{\mathbf{1}},\cdots\boldsymbol{\pi}_{\mathbf{M}},\boldsymbol{\theta},\boldsymbol{\phi},\mathbf{z},\boldsymbol{\sigma},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})\\
&= \left(\prod_{d=1}^{M}\left[\prod_{s=1}^{S_d} p(\mathbf{w}_{\mathbf{d,s}}|\widetilde{\mathbf{z}}_{\mathbf{d,s}},\widetilde{\boldsymbol{\sigma}}_{\mathbf{d,s}},\boldsymbol{\phi}_{\mathbf{z}_{\mathbf{d,s}},\boldsymbol{\sigma}_{\mathbf{d,s}}})\cdot p(\widetilde{\mathbf{z}}_{\mathbf{d,s}}|\boldsymbol{\theta}_{\mathbf{d}})\cdot p(\widetilde{\boldsymbol{\sigma}}_{\mathbf{d,s}}|\boldsymbol{\pi}_{\mathbf{d}})\right] p(\boldsymbol{\theta}_{\mathbf{d}}|\boldsymbol{\alpha})\cdot p(\boldsymbol{\pi}_{\mathbf{d}}|\boldsymbol{\gamma})\right)\left(\prod_{k=1}^{K}\prod_{o=1}^{\Sigma} p(\boldsymbol{\phi}_{\mathbf{k,o}}|\boldsymbol{\beta}_{\mathbf{o}})\right)\\
&\propto \left(\prod_{s=1}^{S_{d'}} p(\mathbf{w}_{\mathbf{d',s}}|\widetilde{\mathbf{z}}_{\mathbf{d',s}},\widetilde{\boldsymbol{\sigma}}_{\mathbf{d',s}},\boldsymbol{\phi}_{(\mathbf{z}_{\mathbf{d'}})_{\mathbf{s'}},(\boldsymbol{\sigma}_{\mathbf{d'}})_{\mathbf{s'}}}) p(\widetilde{\boldsymbol{\sigma}}_{\mathbf{d',s}}|\boldsymbol{\pi}_{\mathbf{d'}})\right) p(\boldsymbol{\pi}_{\mathbf{d'}}|\boldsymbol{\gamma})\\
&\propto \left(\prod_{s=1}^{S_{d'}} p(\widetilde{\boldsymbol{\sigma}}_{\mathbf{d',s}}|\boldsymbol{\pi}_{\mathbf{d'}})\right) p(\boldsymbol{\pi}_{\mathbf{d'}}|\boldsymbol{\gamma})\\
&\propto \prod_{s=1}^{S_{d'}} (\boldsymbol{\pi}_{\mathbf{d'}})_{(\boldsymbol{\sigma}_{\mathbf{d'}})_{s'}} \cdot \prod_{o=1}^{\Sigma} (\boldsymbol{\pi}_{\mathbf{d'}})_o^{(\boldsymbol{\gamma})_o-1}\\
&= \prod_{o=1}^{\Sigma} (\boldsymbol{\pi}_{\mathbf{d'}})_o^{(\mathbf{l}_{\mathbf{d'}})_o+(\boldsymbol{\gamma})_o-1}
\end{aligned} \tag{6.9}$$

Here, $(\mathbf{l}_{\mathbf{d}})_o$ represents the number of times sentiment o is assigned to a sentence in document d. From equation 6.9, it follows that

$$\boldsymbol{\Pi}_{\mathbf{d'}}|\boldsymbol{\theta},\boldsymbol{\pi}_{\mathbf{1}},\ldots\boldsymbol{\pi}_{\mathbf{d'-1}},\boldsymbol{\pi}_{\mathbf{d'+1}},\ldots\boldsymbol{\pi}_{\mathbf{M}},\boldsymbol{\phi},\mathbf{z},\boldsymbol{\sigma},\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma} \sim \text{Dirichlet}(\mathbf{l}_{\mathbf{d'}}+\boldsymbol{\gamma}) \tag{6.10}$$

The last random vector of interest whose conditional distribution need to be determined is $\boldsymbol{\Phi}_{\mathbf{k',o'}}$, the word distribution for topic k' and sentiment o'.

$$\begin{aligned}
p(\boldsymbol{\phi}_{\mathbf{k',o'}}|\boldsymbol{\phi}_{-(\mathbf{k',o'})},\boldsymbol{\theta},\boldsymbol{\pi},\mathbf{z},\boldsymbol{\sigma},\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}) &= \frac{p(\boldsymbol{\phi},\boldsymbol{\theta},\boldsymbol{\pi},\mathbf{z},\boldsymbol{\sigma},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})}{p(\boldsymbol{\phi}_{-(\mathbf{k',o'})},\boldsymbol{\theta},\boldsymbol{\pi},\mathbf{z},\boldsymbol{\sigma},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})}\\
&\propto p(\boldsymbol{\phi},\boldsymbol{\theta},\boldsymbol{\pi},\mathbf{z},\boldsymbol{\sigma},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})\\
&= \left(\prod_{d=1}^{M}\left[\prod_{s=1}^{S_d} p(\mathbf{w}_{\mathbf{d,s}}|\widetilde{\mathbf{z}}_{\mathbf{d,s}},\widetilde{\boldsymbol{\sigma}}_{\mathbf{d,s}},\boldsymbol{\phi}_{\mathbf{z}_{\mathbf{d,s}},\boldsymbol{\sigma}_{\mathbf{d,s}}})\cdot p(\widetilde{\mathbf{z}}_{\mathbf{d,s}}|\boldsymbol{\theta}_{\mathbf{d}})\cdot p(\widetilde{\boldsymbol{\sigma}}_{\mathbf{d,s}}|\boldsymbol{\pi}_{\mathbf{d}})\right] p(\boldsymbol{\theta}_{\mathbf{d}}|\boldsymbol{\alpha})\cdot p(\boldsymbol{\pi}_{\mathbf{d}}|\boldsymbol{\gamma})\right)\left(\prod_{k=1}^{K}\prod_{o=1}^{\Sigma} p(\boldsymbol{\phi}_{\mathbf{k,o}}|\boldsymbol{\beta}_{\mathbf{o}})\right)\\
&\propto \left(\prod_{d=1}^{M}\prod_{s=1}^{S_d} p(\mathbf{w}_{\mathbf{d,s}}|\widetilde{\mathbf{z}}_{\mathbf{d,s}},\widetilde{\boldsymbol{\sigma}}_{\mathbf{d,s}},\boldsymbol{\phi}_{\mathbf{z}_{\mathbf{d,s}},\boldsymbol{\sigma}_{\mathbf{d,s}}})\right)\left(\prod_{k=1}^{K}\prod_{o=1}^{\Sigma} p(\boldsymbol{\phi}_{\mathbf{k,o}}|\boldsymbol{\beta}_{\mathbf{o}})\right)\\
&\propto \left(\prod_{d=1}^{M}\prod_{s=1}^{S_d}\prod_{i=1}^{N_{s'}} (\boldsymbol{\phi}_{\mathbf{z}_{\mathbf{d,s}},\boldsymbol{\sigma}_{\mathbf{d,s}}})_{(\mathbf{w}_{\mathbf{d,s}})_i}\right) p(\boldsymbol{\phi}_{\mathbf{k',o'}}|\boldsymbol{\beta}'_{\mathbf{o}})\\
&\propto \left(\prod_{j=1}^{V} (\boldsymbol{\phi}_{\mathbf{k',o'}})_j^{n_{k',o',j}}\right)\cdot \prod_{j=1}^{V} (\boldsymbol{\phi}_{\mathbf{k',o'}})_j^{(\boldsymbol{\beta}_{\mathbf{o'}})_j-1}\\
&= \prod_{j=1}^{V} (\boldsymbol{\phi}_{\mathbf{k',o'}})_j^{n_{k',o',j}+(\boldsymbol{\beta}_{\mathbf{o'}})_j-1}
\end{aligned} \tag{6.11}$$

Here, $n_{k,s,j}$ represents the number of times word j occurs in a sentence that has topic k and sentiment s. From equation 6.11, it follows, not surprisingly, that also all $\boldsymbol{\Phi}$'s are conditionally Dirichlet distributed.

$$\boldsymbol{\Phi}_{\mathbf{k',o'}}|\boldsymbol{\phi}_{-(\mathbf{k',o'})},\boldsymbol{\theta},\boldsymbol{\pi},\mathbf{z},\boldsymbol{\sigma},\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma} \sim \text{Dirichlet}(\mathbf{n}_{\mathbf{k',o'}}+\boldsymbol{\beta}_{\mathbf{o}}) \tag{6.12}$$

Now also the latent random variables, whose values are not of particular interest to us, need to be sampled. The derivations of their distributions conditional on all other variables can be found below. Note that with $\mathbf{z}$, we mean all topic assignments in the corpus, and with $\mathbf{z}_{-(\mathbf{d',s'})}$ all topic assignments without the topic of sentence s' in document d'.

$$\begin{aligned}
p((\mathbf{z}_{\mathbf{d'}})_{s'}|\mathbf{z}_{-(d',s')},\boldsymbol{\theta},\boldsymbol{\pi},\boldsymbol{\phi},\boldsymbol{\sigma},\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}) &= \frac{p(\mathbf{z},\boldsymbol{\theta},\boldsymbol{\pi},\boldsymbol{\phi},\boldsymbol{\sigma},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})}{p(\mathbf{z}_{-(d',s')},\boldsymbol{\theta},\boldsymbol{\pi},\boldsymbol{\phi},\boldsymbol{\sigma},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})} \\
&\propto p(\mathbf{z},\boldsymbol{\theta},\boldsymbol{\pi},\boldsymbol{\phi},\boldsymbol{\sigma},\mathbf{w}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}) \\
&= \left(\prod_{d=1}^{M}\left[\prod_{s=1}^{S_d} p(\mathbf{w_{d,s}}|\widetilde{\mathbf{z}}_{\mathbf{d,s}},\widetilde{\boldsymbol{\sigma}}_{\mathbf{d,s}},\boldsymbol{\phi}_{\mathbf{z_{d,s}},\boldsymbol{\sigma}_{\mathbf{d,s}}})\cdot p(\widetilde{\mathbf{z}}_{\mathbf{d,s}}|\boldsymbol{\theta}_{\mathbf{d}})\cdot p(\widetilde{\boldsymbol{\sigma}}_{\mathbf{d,s}}|\boldsymbol{\pi}_{\mathbf{d}})\right] p(\boldsymbol{\theta}_{\mathbf{d}}|\boldsymbol{\alpha})\cdot p(\boldsymbol{\pi}_{\mathbf{d}}|\boldsymbol{\gamma})\right)\left(\prod_{k=1}^{K}\prod_{o=1}^{\Sigma} p(\boldsymbol{\phi}_{\mathbf{k,o}}|\boldsymbol{\beta}_{\mathbf{o}})\right) \\
&\propto \prod_{d=1}^{M}\prod_{s=1}^{S_d} p(\mathbf{w_{d,s}}|\widetilde{\mathbf{z}}_{\mathbf{d,s}},\widetilde{\boldsymbol{\sigma}}_{\mathbf{d,s}},\boldsymbol{\phi}_{\mathbf{z_{d,s}},\boldsymbol{\sigma}_{\mathbf{d,s}}})\cdot p(\widetilde{\mathbf{z}}_{\mathbf{d,s}}|\boldsymbol{\theta}_{\mathbf{d}}) \\
&\propto p(\mathbf{w_{d',s'}}|\widetilde{\mathbf{z}}_{\mathbf{d',s'}},\widetilde{\boldsymbol{\sigma}}_{\mathbf{d',s'}},\boldsymbol{\phi}_{(\mathbf{z_{d'}})_{\mathbf{s'}},(\boldsymbol{\sigma}_{\mathbf{d'}})_{\mathbf{s'}}})\cdot p(\widetilde{\mathbf{z}}_{\mathbf{d',s'}}|\boldsymbol{\theta}_{\mathbf{d'}}) \\
&= \left(\prod_{i=1}^{N_{s'}}(\boldsymbol{\phi}_{(\mathbf{z_{d'}})_{\mathbf{s'}},(\boldsymbol{\sigma}_{\mathbf{d'}})_{\mathbf{s'}}})_{(\mathbf{w_{d',s'}})_i}\right)(\boldsymbol{\theta}_{\mathbf{d'}})_{(\mathbf{z_{d'}})_{s'}} \\
&= \left(\prod_{j=1}^{V}(\boldsymbol{\phi}_{(\mathbf{z_{d'}})_{\mathbf{s'}},(\boldsymbol{\sigma}_{\mathbf{d'}})_{\mathbf{s'}}})_j^{n_{(\mathbf{z_{d'}})_{s'},(\boldsymbol{\sigma}_{\mathbf{d'}})_{s'},j}}\right)(\boldsymbol{\theta}_{\mathbf{d'}})_{(\mathbf{z_{d'}})_{s'}}
\end{aligned} \tag{6.13}$$

Therefore, $(\mathbf{Z_{d'}})_{s'}|\mathbf{z}_{-(d',s')},\boldsymbol{\theta},\boldsymbol{\pi},\boldsymbol{\phi},\boldsymbol{\sigma},\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}$ has a Multinomial distribution with parameters:

$$(\widetilde{\mathbf{Z}}_{\mathbf{d',s'}})|\,\text{all other parameters} \sim \text{Multinomial}\left(1,\left[\prod_{j=1}^{V}(\boldsymbol{\phi}_{\mathbf{1},(\boldsymbol{\sigma}_{\mathbf{d'}})_{\mathbf{s'}}})_j^{n_{1,(\boldsymbol{\sigma}_{\mathbf{d'}})_{s'},j}}\right]\cdot(\boldsymbol{\theta}_{\mathbf{d'}})_1,\ldots,\left[\prod_{j=1}^{V}(\boldsymbol{\phi}_{\mathbf{K},(\boldsymbol{\sigma}_{\mathbf{d'}})_{\mathbf{s'}}})_j^{n_{K,(\boldsymbol{\sigma}_{\mathbf{d'}})_{s'},j}}\right]\cdot(\boldsymbol{\theta}_{\mathbf{d'}})_K\right) \tag{6.14}$$

Lastly, the conditional distribution of the sentiment assignment to each phrase in each document is derived.

$$\begin{aligned}
p((\boldsymbol{\sigma}_{\mathbf{d'}})_{s'}|\boldsymbol{\sigma}_{-(d',s')},\boldsymbol{\theta},\boldsymbol{\pi},\boldsymbol{\phi},\mathbf{w},\mathbf{z},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}) &= \frac{p(\boldsymbol{\sigma},\boldsymbol{\theta},\boldsymbol{\pi},\boldsymbol{\phi},\mathbf{w},\mathbf{z}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})}{p(\boldsymbol{\sigma}_{-(d',s')},\boldsymbol{\theta},\boldsymbol{\pi},\boldsymbol{\phi},\mathbf{w},\mathbf{z}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})} \\
&\propto p(\boldsymbol{\sigma},\boldsymbol{\theta},\boldsymbol{\pi},\boldsymbol{\phi},\mathbf{w},\mathbf{z}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}) \\
&= \left(\prod_{d=1}^{M}\left[\prod_{s=1}^{S_d} p(\mathbf{w_{d,s}}|\widetilde{\mathbf{z}}_{\mathbf{d,s}},\widetilde{\boldsymbol{\sigma}}_{\mathbf{d,s}},\boldsymbol{\phi}_{\mathbf{z_{d,s}},\boldsymbol{\sigma}_{\mathbf{d,s}}})\cdot p(\widetilde{\mathbf{z}}_{\mathbf{d,s}}|\boldsymbol{\theta}_{\mathbf{d}})\cdot p(\widetilde{\boldsymbol{\sigma}}_{\mathbf{d,s}}|\boldsymbol{\pi}_{\mathbf{d}})\right] p(\boldsymbol{\theta}_{\mathbf{d}}|\boldsymbol{\alpha})\cdot p(\boldsymbol{\pi}_{\mathbf{d}}|\boldsymbol{\gamma})\right)\left(\prod_{k=1}^{K}\prod_{o=1}^{\Sigma} p(\boldsymbol{\phi}_{\mathbf{k,o}}|\boldsymbol{\beta}_{\mathbf{o}})\right) \\
&\propto \prod_{d=1}^{M}\prod_{s=1}^{S_d} p(\mathbf{w_{d,s}}|\widetilde{\mathbf{z}}_{\mathbf{d,s}},\widetilde{\boldsymbol{\sigma}}_{\mathbf{d,s}},\boldsymbol{\phi}_{\mathbf{z_{d,s}},\boldsymbol{\sigma}_{\mathbf{d,s}}})\cdot p(\widetilde{\boldsymbol{\sigma}}_{\mathbf{d,s}}|\boldsymbol{\pi}_{\mathbf{d}}) \\
&\propto p(\mathbf{w_{d',s'}}|\widetilde{\mathbf{z}}_{\mathbf{d',s'}},\widetilde{\boldsymbol{\sigma}}_{\mathbf{d',s'}},\boldsymbol{\phi}_{(\mathbf{z_{d'}})_{\mathbf{s'}},(\boldsymbol{\sigma}_{\mathbf{d'}})_{\mathbf{s'}}})\cdot p(\widetilde{\boldsymbol{\sigma}}_{\mathbf{d',s'}}|\boldsymbol{\pi}_{\mathbf{d'}}) \\
&= \left(\prod_{i=1}^{N_{s'}}(\boldsymbol{\phi}_{(\mathbf{z_{d'}})_{\mathbf{s'}},(\boldsymbol{\sigma}_{\mathbf{d'}})_{\mathbf{s'}}})_{(\mathbf{w_{d',s'}})_i}\right)(\boldsymbol{\pi}_{\mathbf{d'}})_{(\boldsymbol{\sigma}_{\mathbf{d'}})_{s'}} \\
&= \left(\prod_{j=1}^{V}(\boldsymbol{\phi}_{(\mathbf{z_{d'}})_{\mathbf{s'}},(\boldsymbol{\sigma}_{\mathbf{d'}})_{\mathbf{s'}}})_j^{n_{(\mathbf{z_{d'}})_{s'},(\boldsymbol{\sigma}_{\mathbf{d'}})_{s'},j}}\right)(\boldsymbol{\pi}_{\mathbf{d'}})_{(\boldsymbol{\sigma}_{\mathbf{d'}})_{s'}}
\end{aligned} \tag{6.15}$$

Therefore, $(\boldsymbol{\Sigma}_{\mathbf{d'}})_{s'}|\boldsymbol{\sigma}_{-(d',s')},\boldsymbol{\theta},\boldsymbol{\pi},\boldsymbol{\phi},\mathbf{z},\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}$ has a Multinomial distribution with parameters:

$$(\widetilde{\boldsymbol{\Sigma}}_{\mathbf{d'}})_{s'}|\,\text{all other parameters} \sim \text{Multinomial}\left(1,\left[\prod_{j=1}^{V}(\boldsymbol{\phi}_{(\mathbf{z_{d'}})_{\mathbf{s'}},\mathbf{1}})_j^{n_{(\mathbf{z_{d'}})_{s'},1,j}}\right]\cdot(\boldsymbol{\pi}_{\mathbf{d'}})_1,\ldots,\left[\prod_{j=1}^{V}(\boldsymbol{\phi}_{(\mathbf{z_{d'}})_{\mathbf{s'}},\boldsymbol{\Sigma}})_j^{n_{(\mathbf{z_{d'}})_{s'},\Sigma,j}}\right]\cdot(\boldsymbol{\pi}_{\mathbf{d'}})_\Sigma\right) \tag{6.16}$$

The Gibbs sampling algorithm for the extended version of LDA described in this chapter is given in algorithm 5. Although Gibbs sampling has good convergence properties, it is not implemented in this research, because convergence can take a long time, especially with the many parameter samples that are needed. Programming the algorithm in such a way that its implementation is fast and convergence is reached within a reasonable amount of time, is considered beyond the scope of this thesis.

Algorithm 5 Gibbs Sampling for LDA with syntax and sentiment

```
Initialize $\boldsymbol{\theta_1},\ldots,\boldsymbol{\theta_M},\boldsymbol{\pi_1},\ldots,\boldsymbol{\pi_M},\boldsymbol{\phi_{1,1}},\ldots,\boldsymbol{\phi_{K,\Sigma}},\mathbf{z},\boldsymbol{\sigma}$
Compute initial frequencies $(\mathbf{m_d})_k$ (for $d = 1,\ldots,M$, $k = 1,\ldots,K$), $(\mathbf{l_d})_o$ (for $o = 1,\ldots,\Sigma$)
and $(\mathbf{n_{k,o}})_j$ (for $k = 1$ to $K$,$o = 1,\ldots,\Sigma$ and $j = 1$ to $V$)
Fix $N_{iter}$ for the maximum number of iterations
for $iter = 1$ to $N_{iter}$ do                                      ▷ Sample $N_{iter}$ times
    for $d = 1$ to $M$ do                                          ▷ Iterate over documents
        Draw $\boldsymbol{\Theta_d}$ from Dirichlet($\mathbf{m_d} + \boldsymbol{\alpha}$)
        Draw $\boldsymbol{\Pi_d}$ from Dirichlet($\mathbf{l_d} + \boldsymbol{\gamma}$)
        for $s = 1$ to $S_d$ do                                    ▷ Iterate over phrases
            Draw $\widetilde{\mathbf{Z}}_{\mathbf{d}s}$ from the Multinomial distribution in equation 6.14
            Draw $\widetilde{\boldsymbol{\Sigma}}_{\mathbf{d}s}$ from the Multinomial distribution in equation 6.16
        end for
    end for
    Update frequencies $(\mathbf{n_{k,o}})_j$
    for $k = 1$ to $K$ do                                          ▷ Iterate over topics
        for $o = 1$ to $\Sigma$ do                                 ▷ Iterate over sentiments
            Draw $\boldsymbol{\Phi_{k,o}}$ from Dirichlet($\mathbf{n_{k,o}} + \boldsymbol{\beta_o}$)
        end for
    end for
    Update frequencies $(\mathbf{m_d})_k$ and $(\mathbf{l_d})_o$
end for
Compute posterior estimates of $\boldsymbol{\Theta_1},\ldots,\boldsymbol{\Theta_M},\boldsymbol{\Pi_1},\ldots,\boldsymbol{\Pi_M},\boldsymbol{\Phi_{1,1}},\ldots,\boldsymbol{\Phi_{K,\Sigma}},\mathbf{Z},\boldsymbol{\Sigma}$ using the $N_{iter}$ samples from
their posterior distributions
```

7

Validity of topic-word distribution estimates

In the previous chapters, we have discussed different inference methods for both the plain LDA model and LDA with syntax and sentiment. All methods of inference result in estimates for the latent random variables of interest. That is for each document $d \in \{1, \ldots, M\}$, we want to know the document-topic distribution $\mathbf{\Theta_d}$, and for each topic $k \in \{1, \ldots, K\}$, the topic-word distribution $\mathbf{\Phi_k}$ is estimated.
After having obtained these estimates, we need to take a look at their validity before drawing conclusions. The results are considered not valid, if each topic-word distribution is similar, as will be explained in this chapter. Therefore, the difference in probability vectors is 'measured'.

The most insightful of the latent variables are the topic-word distributions $\mathbf{\Phi}$, from which we can qualitatively see what the topics are about, and therefore what customers find essential to write about in their reviews. Because words form the topics, the human brain can creatively interpret what general theme is behind each topic when looking at, for example, the top 10 words. However, one needs to be careful here, because to be allowed to draw conclusions from the topic-word distribution, also mathematically the topic must be distinct from the others, otherwise it might fit noise or it consists of multiple topics. It can be the case that multiple topics have similar top 10 words, such that these topics are difficult to distinguish. To be able to determine quantitatively which topics are too similar to be interpreted independently, and which are distinctive and unique, different similarity measures are proposed in the literature.

7.1. Normalized symmetric KL-divergence

Koltcov et al. derived a similarity measure based on the Kullback-Leibler divergence. They have found that large proportions of the topics fit noise if the chosen number of topics K is too large. This results in different results for runs with different initializations [24].

Their similarity measure can be thought of as a rescaled symmetric KL-divergence. Symmetric KL-divergence is defined for discrete probability distributions as follows [43].

Definition 7.1 (Symmetric KL-divergence)
The symmetric Kullback-Leibler divergence of a discrete probability distribution $\mathbf{q}$ *with respect to another discrete probability distribution* $\mathbf{p}$*, where* $\mathbf{q}$ *and* $\mathbf{p}$ *have the same support* Ω*, is given by:*

$$KL_{sym}(\mathbf{q}\|\mathbf{p}) = \frac{1}{2}\left(KL(\mathbf{p}\|\mathbf{q}) + KL(\mathbf{q}\|\mathbf{p})\right) \tag{7.1}$$

With:

$$KL(\mathbf{q}\|\mathbf{p}) = \sum_{x\in\Omega} q(x)\log\left(\frac{q(x)}{p(x)}\right) \tag{7.2}$$

Note that the general Kullback-Leibler divergence was already introduced in chapter 2, as the relative entropy. In [24] it is mentioned that the symmetric KL-divergence for the topic distributions of LDA is sensitive to vocabulary sizes because it is dominated by the long tail of rare words in estimate $\boldsymbol{\phi}$. Therefore, one improvement can be to look at, e.g., the top x% words, where the percentage x can be varied and optimized per data set. Another option is to normalize the symmetric KL-divergence to obtain a better interpretable similarity measure. Koltcov et al. introduce the Normalized Kullback-Leibler Similarity (NKLS) measure:

$$NKLS(\mathbf{q}\|\mathbf{p}) = 1 - \frac{KL_{sym}(\mathbf{q}\|\mathbf{p})}{\max_{\mathbf{q}',\mathbf{p}'}\left\{KL_{sym}(\mathbf{q}'\|\mathbf{p}')\right\}} \tag{7.3}$$

The NKLS takes values in the interval $[0,1]$, where 1 is reached if the two probability distributions are exactly equal, and 0 if the two distributions are the most distinctive among all possible combinations of $\mathbf{q}'$ and $\mathbf{p}'$.
In NKLS for LDA, this means that we compare the similarity of each combination of estimated vectors $\boldsymbol{\phi}_k$ and $\boldsymbol{\phi}_l$ (for some $k, l \in \{1,\dots,K\}$) with the two most distinctive vectors among all possible combinations of k and l. The latter gives the maximal KL-divergence of two distributions $\boldsymbol{\phi}$ with respect to each other. A similarity matrix can thus be constructed from which we can conclude which topics are very similar concerning the word probabilities and which topics are more distinctive.
A topic is considered valid if its similarity scores with all other estimated topic-word distributions is larger than a threshold. Koltcov et al. found that a threshold of 0.9 is reasonable, as with this value of NKLS, the top 30-50 words (depending on the size of the vocabulary and the data set) are the same, only the probabilities are different [24]. For values below 0.9, the top 30-50 words can be completely different, while for values above 0.9, the order of the top 30-50 words is almost the same. Therefore, in the results of this thesis, it is also decided to classify topics with a similarity score higher than 0.9 to belong to the same topic or subject, while topics with similarity scores with all other topics below 0.9 are distinctive and can safely be interpreted as results.

Not only for the quality of the topics, the NKLS score can be used, but also to check the stability of the inference procedure for LDA. That is, two runs can be performed with different initializations, and the similarity of the result can be measured. One expects a stable algorithm to give the same topic-word distributions twice. Remember that there is topic exchangeability in LDA, so with the score, we can automatically match the right topic index $k \in \{1,\dots,K\}$ from the first run with index $k' \in \{1,\dots,K\}$ from the second run. In practice, this is too much work in comparison with just sorting the topics based on the average $\bar{\boldsymbol{\theta}}_{\mathbf{M}}$ over all documents for every method. Therefore, the last method is used to ensure that we are comparing results of the same topic permutation.

7.2. Symmetrized Jensen-Shannon divergence

In [43], the symmetrized Jensen-Shannon (JS) divergence is used to determine the similarity between documents, but naturally it can also be applied to find similar topic-word distributions. The JS-divergence is based on the Kullback-Leibler divergence, only it compares a probability distribution with the pointwise arithmetic mean of the same distribution with a second one. In formulas:

$$JS_{sym}(\mathbf{p},\mathbf{q}) = \frac{1}{2}\left[KL\left(\mathbf{p}\|\frac{1}{2}(\mathbf{p}+\mathbf{q})\right) + KL\left(\mathbf{q}\|\frac{1}{2}(\mathbf{p}+\mathbf{q})\right)\right] \tag{7.4}$$

If $\mathbf{p}$ and $\mathbf{q}$ represent the same probability density function, the symmetrized JS-divergence is 0, as the arithmetic mean of the two is equal to both $\mathbf{p}$ and $\mathbf{q}$. According to [43], both the symmetrized JS-divergence as the symmetric KL-divergence work well in practice. Because also for the Jensen-Shannon divergence, it is expected that the long tail of low probabilities will dominate the score, the same type of normalization can be applied, such that we get the normalized Jensen-Shannon similarity measure.

$$NJSS(\mathbf{q}\|\mathbf{p}) = 1 - \frac{JS_{sym}(\mathbf{q}\|\mathbf{p})}{\max_{\mathbf{q}',\mathbf{p}'}\left\{JS_{sym}(\mathbf{q}'\|\mathbf{p}')\right\}} \tag{7.5}$$

One can compute a similarity matrix using either divergence method and compare. A decision on which topics to take into account in the review analysis will be better founded based on both the NKLS and the NJSS.

8
Results

The main research in this thesis can be split up into two different parts, as is represented in the research questions. The first subject concerns the 'basic' model of Latent Dirichlet Allocation and the different inference methods that can be used to estimate the model parameters. The results of this research are given in section 2 of this chapter. However, we need to know more about the properties and shape of the posterior density first, to fully understand the LDA results in section 2. To this end, the first section of this chapter elaborates on the visualization of the posterior density. Secondly, an extension to LDA called 'LDA with syntax and sentiment' has been constructed, whose results on various data sets will be shown in section 3 of this chapter.

8.1. Posterior density visualization of LDA

Before we apply Latent Dirichlet Allocation to actual data sets, it is interesting to learn more about the form of the posterior density. Especially when using the optimization method to find the posterior mode, it is essential to understand its shape.

8.1.1. Influence of the hyperparameters in LDA

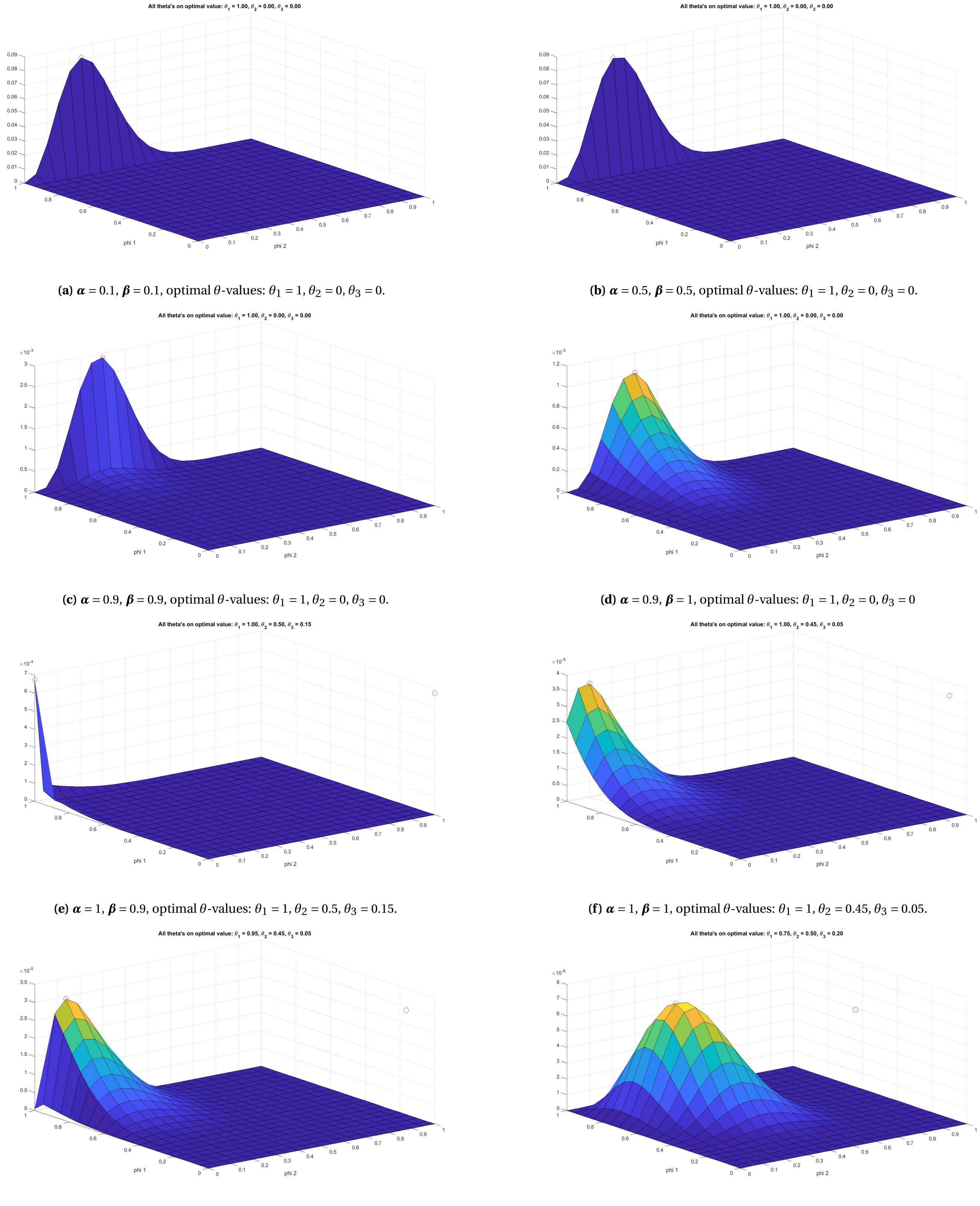
Firstly, we will look at one of the smallest possible data sets to which LDA can be applied. With this example, we want to understand more about the influence of the hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in LDA.
Consider a toy example with one two topics ($K = 2$), two possible words ($V = 2$), and the three documents ($M = 3$):

1. document: [1 1 1 1]
2. document: [1 2 1 1 2 2]
3. document: [1 2 2 2 2 2 2 2]

Remember that the numbers in the document lists stand for either word 1 or word 2. It is not necessary to know what the exact words are to understand this example. The order of the words does not influence the form of the posterior density, since only the frequencies per document are taken into account. Naturally, this is an unrealistic example to apply LDA on, but nevertheless we already have 5 parameters to estimate: θ_1, θ_2, θ_3, ϕ_1 and ϕ_2, making visualization a challenge.

Using a grid on $[\epsilon, 1-\epsilon]^5$ with 21 nodes in each dimension, the posterior density can be computed over the grid. We use ϵ instead of 0 to avoid numerical problems, where ϵ is set to 10^{-8}. Connecting the nodes, we get a 5-dimensional hyperplane that forms the posterior density.

In figure 8.1, the posterior densities are visualized using 8 different settings for hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Because we can only easily understand a three-dimensional surface plot, the conditional densities $p(\phi_1, \phi_2 | \theta_1 = \theta_{1,\text{opt}}, \theta_2 = \theta_{2,\text{opt}}, \theta_3 = \theta_{3,\text{opt}})$ are shown instead of the full posterior densities. With 'opt' is denoted the value of each θ_d (with $d = 1,2,3$) for which the maximum is attained.



(a) $\boldsymbol{\alpha} = 0.1$, $\boldsymbol{\beta} = 0.1$, optimal θ-values: $\theta_1 = 1$, $\theta_2 = 0$, $\theta_3 = 0$.

(b) $\boldsymbol{\alpha} = 0.5$, $\boldsymbol{\beta} = 0.5$, optimal θ-values: $\theta_1 = 1$, $\theta_2 = 0$, $\theta_3 = 0$.

(c) $\boldsymbol{\alpha} = 0.9$, $\boldsymbol{\beta} = 0.9$, optimal θ-values: $\theta_1 = 1$, $\theta_2 = 0$, $\theta_3 = 0$.

(d) $\boldsymbol{\alpha} = 0.9$, $\boldsymbol{\beta} = 1$, optimal θ-values: $\theta_1 = 1$, $\theta_2 = 0$, $\theta_3 = 0$

(e) $\boldsymbol{\alpha} = 1$, $\boldsymbol{\beta} = 0.9$, optimal θ-values: $\theta_1 = 1$, $\theta_2 = 0.5$, $\theta_3 = 0.15$.

(f) $\boldsymbol{\alpha} = 1$, $\boldsymbol{\beta} = 1$, optimal θ-values: $\theta_1 = 1$, $\theta_2 = 0.45$, $\theta_3 = 0.05$.

(g) $\boldsymbol{\alpha} = 1.1$, $\boldsymbol{\beta} = 1.1$, optimal θ-values: $\theta_1 = 0.95$, $\theta_2 = 0.45$, $\theta_3 = 0.05$.

(h) $\boldsymbol{\alpha} = 2$, $\boldsymbol{\beta} = 3$, optimal θ-values: $\theta_1 = 0.75$, $\theta_2 = 0.5$, $\theta_3 = 0.2$

Figure 8.1: Posterior densities for different settings for symmetric hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. There are three documents: $\mathbf{w_1} = [1111]$, $\mathbf{w_2} = [121122]$ and $\mathbf{w_3} = [12222222]$. The vocabulary size is $V = 2$ and the number of topics is $K = 2$. Because the posterior density for this case has 5 parameters and is therefore six-dimensional, the surface plots actually show the joint posterior density of Φ_1 and Φ_2 conditional on the words, hyperparameters and the values of Θ_1, Θ_2 and Θ_3. For the θ_d (with $d = 1,2,3$), the optimal values are taken, that is the values for each θ for which the posterior density is maximal. Note that due to the coarse grid, each value has a rounding error.

The posterior density for this three-dimensional case is given by the the following expression (up to a proportionality constant):

$$p(\theta_1,\theta_3,\theta_3,\phi_1,\phi_2|\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta}) \propto \left[\prod_{d=1}^{3}\prod_{j=1}^{2}\left(\sum_{k=1}^{2}(\boldsymbol{\phi}_\mathbf{k})_j(\boldsymbol{\theta}_\mathbf{d})_k\right)^{(\mathbf{n_d})_j}\right]\cdot\left[\prod_{d=1}^{3}\prod_{k=1}^{2}(\boldsymbol{\theta}_\mathbf{d})_k^{(\boldsymbol{\alpha})_k-1}\right]\cdot\left[\prod_{k=1}^{2}\prod_{j=1}^{2}(\boldsymbol{\phi}_\mathbf{k})_j^{(\boldsymbol{\beta})_j-1}\right] \tag{8.1}$$

Remember that $(\mathbf{n_d})_j$ was defined as the frequency of word j in document d. We can deduce from equation 8.1 that if, for some k, $(\boldsymbol{\alpha})_k < 1$ and $(\boldsymbol{\theta}_\mathbf{d})_k$ is close to 0 or 1, the posterior density will go to $+\infty$. These values for θ are therefore the posterior modes when $(\boldsymbol{\alpha})_k < 1$ for some k, as can be seen in figures 8.1a, 8.1b, 8.1c and 8.1d. The same can be concluded for $\boldsymbol{\beta}$, as shown in figures 8.1a, 8.1b, 8.1c and 8.1e, where the posterior mode estimates for Φ_1 and Φ_2 are either 0 or 1.
For both $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ larger than 1, no numerical problems on the boundaries are found and the posterior mode lies nicely away from the boundaries. Although the conditional posterior density is not a convex plane (strictly speaking) the posterior mode can be easily found using optimization methods, see for example figure 8.1h.

8.1.2. VBEM's posterior density approximation

VBEM's approximation of the posterior density can be visualized in the same way. In the Variational Bayesian Expectation-Maximization method, we use auxiliary functions with variational parameters $\boldsymbol{\gamma}_\mathbf{1},\boldsymbol{\gamma}_\mathbf{2},\boldsymbol{\gamma}_\mathbf{3},\boldsymbol{\phi}_\mathbf{1}$ and $\boldsymbol{\phi}_\mathbf{2}$, such that the approximation function q for the posterior density is given by:

$$\begin{aligned} q(\theta_1,\theta_2,\theta_3,\phi_1,\phi_2) &= \left(\prod_{d=1}^{3} q(\boldsymbol{\theta}_\mathbf{d};\boldsymbol{\gamma}_\mathbf{d})\right)\left(\prod_{j=1}^{2} q(\boldsymbol{\phi}_\mathbf{j};\boldsymbol{\lambda}_\mathbf{j})\right) \\ &= \left(\prod_{d=1}^{3}\left[\frac{\Gamma(\sum_{k=1}^{2}(\boldsymbol{\gamma}_\mathbf{d})_k)}{\prod_{k=1}^{2}\Gamma((\boldsymbol{\gamma}_\mathbf{d})_k)}\right](\boldsymbol{\theta}_\mathbf{d})_1^{(\boldsymbol{\gamma}_\mathbf{d})_1-1}\cdot(1-(\boldsymbol{\theta}_\mathbf{d})_1)^{(\boldsymbol{\gamma}_\mathbf{d})_2-1}\right) \\ &\quad\cdot\left(\prod_{k=1}^{2}\left[\frac{\Gamma(\sum_{j=1}^{2}(\boldsymbol{\lambda}_\mathbf{k})_j)}{\prod_{j=1}^{2}\Gamma((\boldsymbol{\lambda}_\mathbf{k})_j)}\right](\boldsymbol{\phi}_\mathbf{k})_1^{(\boldsymbol{\lambda}_\mathbf{k})_1-1}\cdot(1-(\boldsymbol{\phi}_\mathbf{k})_1)^{(\boldsymbol{\lambda}_\mathbf{k})_2-1}\right) \end{aligned} \tag{8.2}$$

With the variational parameter vectors $\boldsymbol{\gamma}_\mathbf{1},\boldsymbol{\gamma}_\mathbf{2},\boldsymbol{\gamma}_\mathbf{3},\boldsymbol{\phi}_\mathbf{1},\boldsymbol{\phi}_\mathbf{2}$ determined such that q approximates the posterior density as well as possible, we can compute the posterior mode of q. Because the data set in this example is small, this can be done using grid search over the relatively coarse grid. Naturally, we are limited by the grid size, but an approximation of the maximum can be found. It is not likely that there exists a value between two nodes that is much higher than the values of q on these same two nodes.
A plot with both the true posterior density and the approximate posterior density q is given in figure 8.2. We have fixed θ_1, θ_2 and θ_3 on their posterior mode values, since we can only plot a three-dimensional graph.

We are only interested in the location of the posterior mode, not the posterior mode itself. We see in figure 8.2 that the locations of the maximum of the posterior density and the approximate density are not the same. This is caused by the mean-field approximation in the approximation function.
In section 4.2.1, the posterior density for all latent variables, thus including topic assignments $\mathbf{Z}$, is approximated with $q(\boldsymbol{\theta},\boldsymbol{\phi},\mathbf{z})$. However, we are only interested in the posterior mode estimates of document-topic distributions $\boldsymbol{\Theta}$ and the topic-word distributions $\boldsymbol{\Phi}$. Therefore, in this example, the topic assignments $\mathbf{Z}$ are integrated out. Because of the mean-field approximation, we can easily see that:

$$\sum_{\mathbf{z}} q(\boldsymbol{\theta},\boldsymbol{\phi},\mathbf{z}) = q(\boldsymbol{\theta},\boldsymbol{\phi})\cdot\sum_{\mathbf{z}} q(\mathbf{z}) = q(\boldsymbol{\theta},\boldsymbol{\phi})\cdot 1 \tag{8.3}$$

since each auxiliary function q is a probability density. Therefore, after having determined $q(\boldsymbol{\theta},\boldsymbol{\phi},\mathbf{z})$ using VBEM, we can just ignore $q(\mathbf{z})$.

For the data in this example, there are two posterior modes due to topic exchangeability. The posterior modes of the true posterior density found using grid search are: $\theta_1 = 0.5$, $\theta_2 = 0.8$, $\theta_3 = 0.25$, $\phi_1 = 1$, $\phi_2 = 0$ and $\theta_1 = 0.5$, $\theta_2 = 0.2$, $\theta_3 = 0.75$, $\phi_1 = 0$, $\phi_2 = 1$. The first mode is shown in figure 8.2. The location of the maximum of the approximate posterior density from VBEM is: $\theta_1 = 0.5$, $\theta_2 = 0.75$, $\theta_3 = 0.3$, $\phi_1 = 0.75$, $\phi_2 = 0.15$. It is clear that the approximation function aimed to approximate the posterior density in its first posterior mode, but the model parameter estimates differ quite a lot. Therefore, we conclude that the VBEM algorithm can approximate the posterior density in a general good direction, but the approximation is still too far off to draw conclusions for the model parameters. This conclusion will be supported with the application of LDA to two data sets.

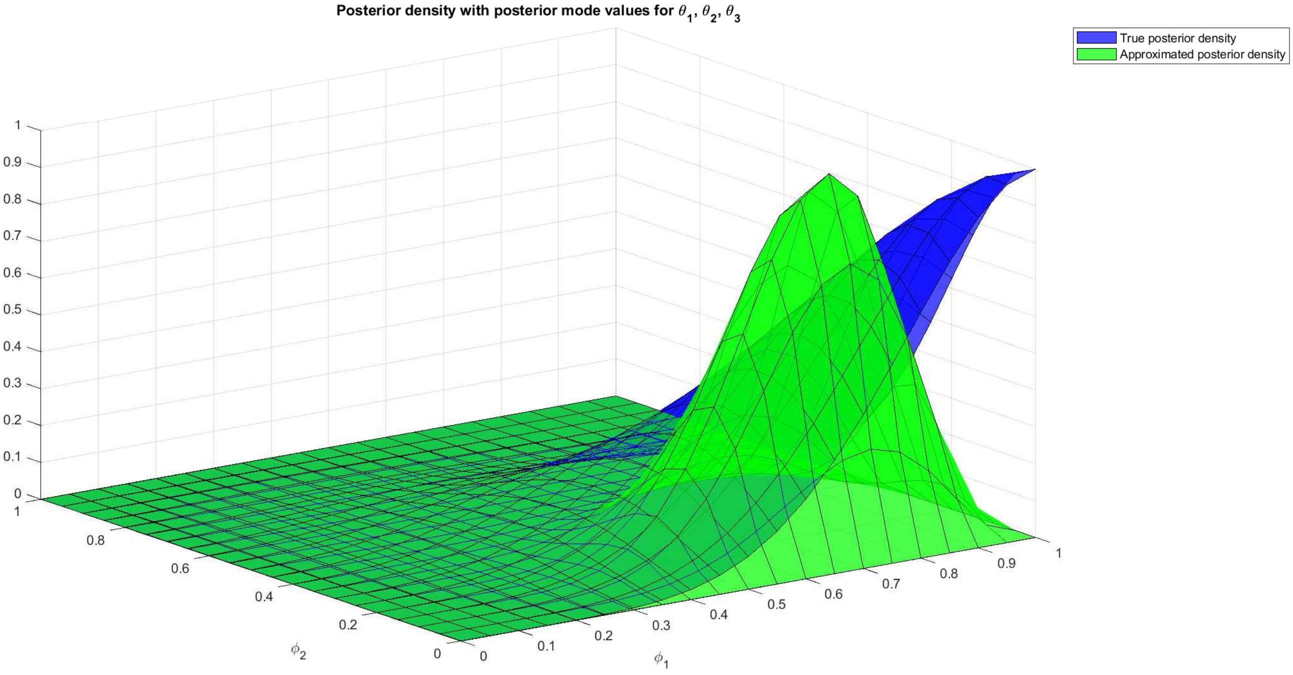


Figure 8.2: Surface plot of join posterior density of Φ_1, Φ_2 conditional on $\Theta_1 = 0.5$, $\Theta_2 = 0.8$, $\Theta_3 = 0.25$. Hyperparameters are set to $\boldsymbol{\alpha} = 1.1$ and $\boldsymbol{\beta} = 1$. The number of topics is $K = 2$, the vocabulary size is $V = 2$ and there are 3 documents: $M = 3$. The data consists of three documents: $\mathbf{w_1} = [1,2]$, $\mathbf{w_2} = [1,1,1,1,2]$ and $\mathbf{w_3} = [1,2,2,2,1,2,2,2]$. Also the approximation of the conditional posterior density, q, is shown and it can be seen that their maxima lie on different locations, resulting in different posterior modes. Note that for the sake of comparison, all values are normalized such that the maximal value equal 1 for both surfaces.

8.2. LDA: different methods of inference

There are different methods to estimate the model parameters[1] $\boldsymbol{\Theta_d}$ with $d = 1, \dots, M$, and $\boldsymbol{\Phi_k}$ with $k = 1, \dots, K$, respectively the document-topic distributions and the topic-word distributions. The estimators are based on the posterior mean or the posterior mode.

Although the posterior mean estimator calculated from the complete posterior distribution does not result in good explanatory results for the topic and word distribution due to topic exchangeability, it is used in Gibbs sampling. This is possible because the Gibbs sampling algorithm is expected to 'circle' around one topic permutation, as the probability to go from one hill in the posterior density to another is very small. Therefore, we use the fact that Gibbs sampling does not work properly (in terms of convergence), to get informative estimators for our latent random variables of interest. There is a good implementation of Gibbs sampling for LDA in the open source program KNIME [23]. The results from this implementation are considered to be good Gibbs sampling results, even though it is not entirely clear what steps are taken in this software exactly. In the documentation, methods described in [32, 52] are referred to.

Another possibility, apart from programming the Gibbs algorithm with the update formulas from algorithm 1 yourself, is using the *JAGS* package in either R or Python, whichever is preferred. *JAGS* stands for 'just another Gibbs sampler' and allows for Markov chain Monte Carlo sampling for almost every hierarchical Bayesian model. Only the conditional distributions need to be specified, then *JAGS* determines whether Gibbs sampling can be done or Metropolis-Hastings sampling is needed. Remember that Gibbs sampling is only possible if the

[1] Strictly speaking, these are latent random variables in a Bayesian setting.

distributions and dependencies in the hierarchical model are chosen such that the conditional distribution of each parameter given all other parameters and data is of a known, closed form. Unfortunately, *JAGS* is not a very fast implementation of Gibbs. Therefore, results are not generated by this implementation.

The posterior mode is very challenging to compute analytically. However, we can use the posterior distribution in its exact form and look for an optimum. This optimization method is described in chapter 5. Unfortunately, it cannot be guaranteed that a global optimum is found, only local optima can be reached by the algorithm. Besides, there are many local optima due to topic exchangeability. We conclude that all posterior modes that are symmetric due to topic exchangeability are equally good and give the same result concerning how the topics are distributed and what words are most frequently used per topic (i.e., estimations for respectively $\mathbf{\Theta}$ and $\mathbf{\Phi}$), based on observations from low-dimensional versions of LDA. The optimization algorithm is not stable in finding the same posterior mode every time, as different initializations are used. However, when the topics are sorted in the same way after every optimization, we can conclude that indeed, every posterior mode is equally good and give the same results when looking at the estimations of $\mathbf{\Theta}$ and $\mathbf{\Phi}$.
Another method to estimate model parameters via the posterior model that is often used by topic modelers is Variational Bayesian Expectation Maximization (VBEM). In particular, VBEM with the mean-field approximation is very common. In this method, the posterior distribution is approximated by a simpler function of which the mode can be computed analytically. The approximation function is based on the mean-field approximation, which means that each latent variable is assumed to be independent, such that the product of their marginal density functions gives an approximation of the posterior distribution. Naturally, this is a very strong assumption, but we will see in the results that the method can perform relatively well in terms of the estimations for the document-topic and topic-word distributions.

8.2.1. Small data set: Cats and Dogs

First, a simple data set with distinctive documents is considered. This data set is created by the author of this thesis, based on the principle that if two documents tell about a different subject and use different words to do that, these documents belong to two distinct topics. The data set and the preprocessed version of it can be found in appendix B.3. There are documents about dogs, and documents about cats. Also, some texts write about animals in general, but the words used in these documents are also used in the cat documents, meaning that they are expected to be assigned to the 'cat'-topic. If a person who understands English divided the documents into two clusters or topics, this person would get the following classification.

Table 8.1: 'Cats and dogs' data set. Simple, small data set with distinctive topic clusters by construction. Documents in gray belong to the 'cat' topic, while those in white are part of the 'dog' topic.

Documents
cats are animals
dogs are canids
cats are fluffy
dogs bark
cats meow fluffy are cats animals are large
dogs bite
cats scratch
dogs bite
cats scratch
dogs bark
cats are fluffy animals are cool not all animals are fluffy
dogs are tough canids are special bark dogs
cool cats

Then, the topic-word distributions would be:

Table 8.2: Estimates of the topic-word distributions $\mathbf{\Phi_1}$ and $\mathbf{\Phi_2}$, based on the intuitive construction of topics by reading the documents. The probabilities are calculated using relative frequencies.

Topic 1: $\hat{\boldsymbol{\phi}}_1$		**Topic 2: $\hat{\boldsymbol{\phi}}_2$**	
Words	Probabilities	Words	Probabilities
cats	0.364	dogs	0.438
fluffy	0.182	bark	0.188
animals	0.182	bite	0.125
cool	0.0909	canids	0.125
scratch	0.0909	special	0.0625
large	0.0455	tough	0.0625
meow	0.0455		

The word lists are the words that occur in documents belonging to either topic 1 or topic 2. Consequently, probabilities are computed by taking the relative frequencies, that is, the frequency of a word in all documents belonging to a particular topic, divided by the total number of words in all documents that are assigned to that topic.

For this small data set, it is possible to read all documents and assign them to a topic, especially when there are only two topics. However, in case of multiple topics within one document, this becomes more difficult. Also, the aim of the conducted research in this thesis is to do unsupervised Latent Dirichlet Allocation. We want to avoid reading reviews, and rather let the algorithm decide what topics are hidden in the data set.
Therefore, three algorithms are run to find the topics in the 'Cats and dogs' data set: Gibbs sampling using KNIME, Variational Bayesian EM using Python's *gensim* package, and Adam optimization to find the posterior mode. It is already known that there are only two topics, so $K = 2$. Then, hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ need to be chosen. For simplicity, we take symmetric priors, that is $(\boldsymbol{\alpha})_i = (\boldsymbol{\alpha})_j$ for all $i, j = 1, \ldots, K$ and similarly for $\boldsymbol{\beta}$. Different combinations of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are used, and estimates for $\mathbf{\Theta}$ and $\mathbf{\Phi}$ are determined for each setting and each method.

One of the settings in which the topic assignment per document corresponds to those in table 8.1 is $(\boldsymbol{\alpha})_i = 0.99$ and $(\boldsymbol{\beta})_i = 1$. In the tables below, the estimates for $\mathbf{\Phi_1}$ and $\mathbf{\Phi_2}$ per inference method for LDA are shown.

Table 8.3: Optimization results for $\hat{\boldsymbol{\phi}}_1$ and $\hat{\boldsymbol{\phi}}_2$ with symmetric $(\boldsymbol{\alpha})_i = 0.99$ and symmetric $(\boldsymbol{\beta})_i = 1$. The Adam gradient descent algorithm is used with a learning rate of 0.001, a stopping criterion of 10^{-4} for the relative change and a maximum of 20,000 iterations. Regularization as described in 5.4.2 is applied.

Topic 1: $\hat{\boldsymbol{\phi}}_1$		**Topic 2: $\hat{\boldsymbol{\phi}}_2$**	
Words	Probabilities	Words	Probabilities
dogs	0.437	cats	0.364
bark	0.187	fluffy	0.182
canids	0.124	animals	0.182
bite	0.124	cool	0.0909
special	0.0625	scratch	0.0909
tough	0.0625	meow	0.0455
animals	3.11E-05	large	0.0454
cats	1.29E-06	dogs	3.48E-05
meow	1.08E-06	special	3.27E-05
fluffy	8.82E-07	bite	2.51E-05
cool	6.42E-07	tough	1.44E-05
large	4.87E-07	bark	1.25E-05
scratch	3.41E-07	canids	7.64E-06

Table 8.4: Variational Bayesian EM results for $\mathbf{\Phi_1}$ and $\mathbf{\Phi_2}$ with symmetric $(\boldsymbol{\alpha})_i = 0.99$ and symmetric $(\boldsymbol{\beta})_i = 1$.

Topic 1: $\hat{\boldsymbol{\phi}}_1$		**Topic 2: $\hat{\boldsymbol{\phi}}_2$**	
Words	Probabilities	Words	Probabilities
dogs	0.307	cats	0.283
bark	0.144	animals	0.160
bite	0.103	fluffy	0.154
canids	0.102	cool	0.0861
special	0.0615	scratch	0.0754
tough	0.0557	large	0.0523
cats	0.0520	meow	0.0508
scratch	0.0394	dogs	0.0278
fluffy	0.0319	tough	0.0250
cool	0.0269	bark	0.0222
animals	0.0256	canids	0.0216
meow	0.0255	bite	0.0206
large	0.0238	special	0.0201

Table 8.5: Gibbs sampling results from KNIME for $\mathbf{\Phi_1}$ and $\mathbf{\Phi_2}$ with symmetric $(\boldsymbol{\alpha})_i = 0.99$ and symmetric $(\boldsymbol{\beta})_i = 1$. 1000 iterations are executed on 8 different threads.

Topic 1: $\hat{\boldsymbol{\phi}}_1$		**Topic 2: $\hat{\boldsymbol{\phi}}_2$**	
Words	Probabilities	Words	Probabilities
dogs	0.438	cats	0.364
bark	0.188	animals	0.182
bite	0.125	fluffy	0.182
canids	0.125	cool	0.0909
special	0.0625	scratch	0.0909
tough	0.0625	large	0.0455
animals	0	meow	0.0455
cats	0	bark	0
cool	0	bite	0
fluffy	0	canids	0
large	0	dogs	0
meow	0	special	0
scratch	0	tough	0

Note that the Gibbs sampling results in table 8.5 correspond precisely to those intuitively constructed in table 8.2. The posterior mode estimates for $\mathbf{\Phi_1}$ and $\mathbf{\Phi_2}$ via optimization in table 8.3 are also very similar to the intuitive result, only a small probability is assigned to the words that do not actually belong to that topic. The fact that the small probabilities are not 0 is one of the properties of the optimization algorithm, caused by the regularization term.
The Variational Bayesian EM algorithm does find the right topic assignment for each document and the right top words for each topic, but there is still some probability mass left for words that do not belong to, for example, the cat topic. This already shows the lack of accuracy of this algorithm, since in this simple case with few documents and a very clear distinction between documents, the performance is still not optimal. However, the results from the VBEM algorithm can be used as input for the posterior mode optimization algorithm, namely as initial condition. It is found that this significantly reduces the number of iterations needed to find the posterior mode, while the same results as in table 8.3 are obtained.

In total, 18 different combinations for symmetric $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are taken. Note that with $(\boldsymbol{\alpha})_i = 1$, we mean that each element of vector $\boldsymbol{\alpha}$ equals 1.
First, $(\boldsymbol{\beta})_i$ is kept constant at 1, while $(\boldsymbol{\alpha})_i$ took values: $(\boldsymbol{\alpha})_i = 0.25, 0.5, 0.75, 0.9, 0.99, 1, 1.5, 2, 5$. Within these settings $(\boldsymbol{\alpha})_i = 0.99$ showed the best results, therefore in the second sweep $(\boldsymbol{\alpha})_i$ is kept constant at $(\boldsymbol{\alpha})_i = 0.99$, while $(\boldsymbol{\beta})_i = 0.25, 0.5, 0.75, 0.9, 0.99, 1, 1.5, 2, 5$. Note that the $(\boldsymbol{\beta})_i$ hyperparameter cannot be altered in the VBEM algorithm from *gensim*. The settings for $(\boldsymbol{\alpha})_i$ and $(\boldsymbol{\beta})_i$ in which the estimates for both $\mathbf{\Theta_d}$ for $d = 1, \ldots, M$, and $\mathbf{\Phi_1}$ and $\mathbf{\Phi_2}$ are correspondent with the intuitive results in terms of order of magnitude of the document-topic and topic-word probabilities, are given in the table below.

Table 8.6: Combinations of hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, both taken as symmetric vectors, for which the optimization results are satisfactory concerning estimations of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$.

$(\boldsymbol{\alpha})_i$	$(\boldsymbol{\beta})_i$
0.99	1
2	1
0.99	0.99
0.99	2
0.99	5

In general, when choosing the hyperparameters, you need to think about what you expect from the documents. If you expect that there is only one topic per document, $(\boldsymbol{\alpha})_i$ needs to be smaller than 1. From the Dirichlet distribution, it is known that the smaller $(\boldsymbol{\alpha})_i$, the more likely it is to draw a distribution that is almost a unit vector. On the other hand, if $(\boldsymbol{\alpha})_i$ is larger than 1, a distribution with equal probabilities for each dimension is preferred. When $(\boldsymbol{\alpha})_i = 1$, you do not know anything about the topic distribution per document. It can be about only one topic or about K topics. The data will guide you towards good estimations for topic distribution $\boldsymbol{\theta_d}$ for each document d. The same mechanism applies to the hyperparameter $(\boldsymbol{\beta})_i$. Because a topic is, in general, about more than one word, a small $(\boldsymbol{\beta})_i$ is not wise to take. Some topics are expected to be about a few words, while other topics can write about a large list of words, or they are a 'noise' topic. That is a topic to which all documents or words in documents that cannot directly be assigned to a specific subject, are assigned. Think about background words, or simply stories people tell in a review that are so unique that they do not form a topic. If both $(\boldsymbol{\alpha})_i$ and $(\boldsymbol{\beta})_i$ are 1, the posterior density is only proportional to the likelihood, such that posterior mode estimation actually becomes maximum likelihood estimation.
The optimization algorithm does not handle small $(\boldsymbol{\alpha})_i$ or $(\boldsymbol{\beta})_i$ well because these values result in a $-\log(\text{posterior})$ of $-\infty$ when values of $(\mathbf{\Theta_d})_i$ from some d and i, or $(\mathbf{\Phi_k})_j$ from some k and j are close to 0. Once the optimization algorithm steps towards these boundaries, it will only push the value of the small-valued parameter further towards 0, as this value minimizes the $-\log(\text{posterior})$. Therefore, it is better to take $(\boldsymbol{\alpha})_i$ or $(\boldsymbol{\beta})_i$ close to 1, but only a little smaller, like 0.99. In this way, the mechanism of only one topic per document or a preference for only a small number of words per topic is maintained, but the optimization algorithm 'falls' less quickly into the abyss at the boundaries.

Naturally, the 'Cats and Dogs' data set was a simple example that is not representative of the use of LDA in practice. Therefore, a real review data set is taken, and the same analyses are done. Only now, it is already known that we should not take $(\boldsymbol{\alpha})_i$ and $(\boldsymbol{\beta})_i$ too small.

8.2.2. Towards more realistic analyses: stroller data

The stroller data set consists of 2000 reviews from Amazon concerning different brands and types of strollers. The goal is to retrieve about which aspects, issues or stories people write in reviews. We expect some elements of a stroller to each form a topic, but also a topic with only positive or negative opinion words, without many explanatory words, think of for example *'Great!'* or *'I love it'.* Furthermore, there can be some reviews with spam or advertisements included in the data set. These documents are also expected to form a topic. With this reasoning, we set the number of topics to 10: $K = 10$. The reviews are relatively large (sizes vary between 1 and 500 words), so we want $(\boldsymbol{\alpha})_i$ to be close to 1. In this setting, a document/review can write about only one topic, but also about five topics for example. Variation among document-topic distribution is still possible, as $(\boldsymbol{\alpha})_i$ is close to 1. Beforehand, we believe that there will be a slight preference for a few topics in a document. It is not likely that a review writes about all ten topics. Therefore $(\boldsymbol{\alpha})_i$ slightly smaller than 1 is expected to give the best results.

Three different methods of inference are used: Gibbs sampling using KNIME, Variational Bayesian EM using the *gensim* package in Python and Adam optimization for finding the posterior mode. The maximal vocabulary size V is set to 2000, meaning that if there are more than 2000 different words in all 2000 reviews together, those that occur the least frequently are removed.
Note that, in total, $K \cdot M$ (document-topic probabilities) plus $K \cdot V$ (topic-word probabilities) parameters need to be estimated, resulting in at most 40000 (taking $V = 2000$) estimations. This parameter space is enormous, resulting in slow convergence in both the Gibbs and VBEM algorithms. Also, the optimization method has more problems with finding the optimum, especially when either $(\boldsymbol{\alpha})_i$ or $(\boldsymbol{\beta})_i$ is smaller than 1. Therefore, the regularization term is given more weight to keep the parameter estimations away from the boundaries of the domain $[0,1]$.

Because it is not easily feasible to read all reviews and manually assign topics to them, we cannot compare the inference results with natural results, as had been done for the 'Cats and Dogs' data set. Therefore, model validation measures come into place. In section 2.4, perplexity was introduced for a general training and test set. To compute the perplexity for the LDA model, we need to define it further. For a test set consisting of M documents, each having N_d words, document-topic distribution $\boldsymbol{\theta_d}$ and given the topic-word distributions $\boldsymbol{\phi_k}$ for $k = 1, \ldots, K$, the perplexity is computed using:

$$\begin{aligned}\text{Perplexity}(\mathbf{w_{test}}) &= \exp\left\{-\frac{\log(\mathbb{P}(\mathbf{w_{test}}|\boldsymbol{\theta},\boldsymbol{\phi}))}{|\mathbf{w_{test}}|}\right\} \\ &\propto \exp\left\{-\frac{\sum_{d=1}^{M}\sum_{j=1}^{V}(\mathbf{n_d})_j \log\left(\sum_{k=1}^{K}(\boldsymbol{\phi_k})_j \cdot (\boldsymbol{\theta_d})_k\right)}{\sum_{d=1}^{M} N_d}\right\}\end{aligned} \tag{8.4}$$

The perplexity compares the inferred model with the case in which each word is equally likely, which is the least informative model and has the highest entropy. The lower the perplexity, the better the model, that is, the more informative the model has retrieved from the data set.
The computation of the perplexity for our text data is not straightforward. Every document d has its own estimated parameter vector $\boldsymbol{\theta_d}$. The exact meaning and independence assumptions of all parameters in Latent Dirichlet Allocation will be elaborated on in the next chapter. For now, it is important to understand that we cannot split the review data set into a set of reviews belonging to the training data set, and a set of reviews belonging to the test set. Namely, the parameter vectors $\boldsymbol{\theta_d}$ for all documents d in the test set cannot be estimated. Therefore, the data set is split into a train and test set differently. Every document consists of a set of words, which can easily be split into two. The largest part, in this thesis 80% of the words (up to a rounding error), is assigned to the training set. The remaining part of the document is then the test set. With this setting, all model parameters are estimated using the training set, and perplexity is calculated with the test set. This method to compute the perplexity is proposed in [48].
Another way of comparing inferred parameters can be done by looking at the logarithm of the posterior, which resembles maximum likelihood estimation, only now, instead of looking at the likelihood, we use the posterior distribution. That is the likelihood times the prior distributions. Earlier in this thesis, Bayesian statistics was introduced. Here we made a distinction between posterior mean and posterior mode estimator. If different methods are used to estimate the posterior mode, it is straightforward to substitute these parameter estimations into the posterior distribution as a check which results in the highest posterior, or highest log posterior. Therefore, the value of the log posterior is also used for model comparison. The model with the highest log posterior value is considered the best.

On a server with a fast GPU (Graphical Processor Unit), the optimization and Variational Bayesian EM have been run. The optimization algorithm can be run parallel using *Tensorflow*, such that a lot of time is gained when running the program on a fast GPU. Again, different values for $(\boldsymbol{\alpha})_i$ and $(\boldsymbol{\beta})_i$ are taken and summarized in table 8.7.

Table 8.7: Overview of results for LDA on stroller data using different inference methods. Adam optimization and Variational Bayesian Expectation Maximization are used to determine the posterior mode, while the KNIME implementation estimates the model parameters via Gibbs sampling. Two different initialization methods are used for the optimization algorithm: random initialization and taking the estimates of VBEM as the initial value. Besides the model validation scores, it is an indication whether, for either optimization method, the maximum number of iterations is reached. The optimization method is truncated after $\frac{100}{\text{learning rate}}$ iterations.

		Perplexity				Log posterior				Truncation
$(\boldsymbol{\alpha})_i$	$(\boldsymbol{\beta})_i$	Optim random init.	Optim VBEM init.	KNIME	VBEM	Optim random init.	Optim VBEM init.	KNIME	VBEM	Optim
0.8	0.1	$6.43 \cdot 10^9$	$5.54 \cdot 10^{8*}$	691	689^{**}	$1.00 \cdot 10^5$	$1.27 \cdot 10^{5*}$	$1.12 \cdot 10^5$	inf**	x
0.9	0.9	$1.76 \cdot 10^8$	$8.46 \cdot 10^{7*}$	695	681^{**}	$-3.14 \cdot 10^5$	$-3.24 \cdot 10^{5*}$	$-2.85 \cdot 10^5$	inf**	x
0.99	0.99	1279	1019^*	686	675^{**}	$-3.96 \cdot 10^5$	$-4.23 \cdot 10^5$ *	$-3.35 \cdot 10^5$	inf**	
0.999	0.999	1032	994^*	461	491^{**}	$-4.05 \cdot 10^5$	$-4.33 \cdot 10^{5*}$	$-3.40 \cdot 10^5$	inf **	
0.1	1	2072	2512	717	847	$7.35 \cdot 10^5$	$1.03 \cdot 10^6$	$-2.64 \cdot 10^5$	inf	x
1	0.01	$4.01 \cdot 10^5$	7644^*	678	675^{**}	$-2.86 \cdot 10^5$	$-1.04 \cdot 10^{5*}$	$1.14 \cdot 10^5$	NaN**	x
1.5	0.9	761	753^*	668	665^{**}	$-3.55 \cdot 10^5$	$-4.04 \cdot 10^{5*}$	$-3.27 \cdot 10^5$	-inf**	x
1	1	1286	1031	692	675	$-4.03 \cdot 10^5$	$-4.35 \cdot 10^5$	$-3.41 \cdot 10^5$	NaN	
1.1	1.1	821	850^*	913	668^{**}	$-4.05 \cdot 10^5$	$-4.64 \cdot 10^{5*}$	$-4.42 \cdot 10^5$	-inf**	

In the VBEM application in the python package *gensim*, parameter $(\boldsymbol{\beta})_i$ cannot be tuned. Therefore, at $(*)$, we used the results of VBEM with the corresponding value for $(\boldsymbol{\alpha})_i$ and $(\boldsymbol{\beta})_i = 1$ as initialization for Adam optimization. At $(**)$, the model validation scores are actually given for the corresponding $(\boldsymbol{\alpha})_i$ and $(\boldsymbol{\beta})_i = 1$, so one needs to be careful with comparing the methods with each other.
Whether or not truncation is applied in the optimization method is given for both types of initialization. For each setting of the hyperparameters for which the maximum number of iterations was reached in the optimization method, truncation took place for both optimization with random initialization and with VBEM initialization. This means that the algorithm did not converge faster when using VBEM initialization with this stopping criterion.

In table 8.7, we see many effects and peculiarities. First of all, the perplexity blows up for the results of the inferred parameters using Adam optimization and with both $(\boldsymbol{\alpha})_i$ and $(\boldsymbol{\beta})_i$ smaller than 1. This is expected, as the optimization algorithm pushes parameter values of most dimensions in the direction of 0, except a few, such that still for every $\boldsymbol{\theta}_\mathbf{d}$ and $\boldsymbol{\phi}_\mathbf{k}$ for $d = 1, \dots M$ and $k = 1, \dots K$, the sum is equal to one. If then, in the test set occurs a word that has a very small probability to be in that document, when looking at the combination of $\hat{\boldsymbol{\theta}}_\mathbf{d}$ and the topic-word probabilities $\hat{\boldsymbol{\phi}}_\mathbf{k}$ for $k = 1, \dots, K$, the contribution of that element's probability to the perplexity is enormous. Remember the definition of perplexity in equation 8.4. If, in some document, word j does occur in the test set (i.e. $(\mathbf{n_d})_j > 0$), but its probability in all $\hat{\boldsymbol{\phi}}_\mathbf{k}$ is very small, the log of something close to zero is a large negative number. This results in a large value of the perplexity.

The results using Gibbs sampling by KNIME do not suffer from this effect. That is Gibbs sampling always assigns some weight to a word. In at least one topic-word distribution estimate $\boldsymbol{\Phi}_\mathbf{k}$ (for $k = 1, \dots, K$) from KNIME, the word probability is larger than 10^{-6}. Contrary to the optimization results, where the smallest word probability summed over all topics is of the order of magnitude 10^{-17}. This declares the large difference in perplexity values between the optimization method and KNIME. Therefore, we can conclude that Gibbs sampling has better predictive performance than the posterior mode estimates via Adam optimization.

The only setting in which we can compare VBEMs perplexity score with the other ones, and in which VBEM performs best is for $(\boldsymbol{\alpha})_i = 1$ and $(\boldsymbol{\beta})_i = 1$. Remember that the lower the perplexity, the better the model, as explained in section 2.4. When a closer look is taken at the actual parameter estimates by VBEM, we see that the document-topic and topic-word distributions are relatively flat. This phenomenon could also be observed in table 8.4, where all words are given a relatively large probability, and the highest probability for the top word ‘dogs’ is only 0.307, while this word receives a probability of 0.438 in estimates by the Gibbs sampling method. Naturally, when all words are given a relatively large probability in at least one topic-word distribution, the predictive performance is higher. However, the aim of the application of LDA in marketing intelligence is to describe and summarize the considered data set, not to predict what the topics in the next review will be.

The second model comparison measure that is reported in table 8.7 is the log posterior. This is just the natural logarithm of the posterior distribution, in which all constants are left out. Because these constants are the same for each inference method, their omission has no influence on model comparison. Remember the log posterior distribution for general LDA:

$$\log(\text{posterior}) = C + \sum_{d=1}^{M}\sum_{j=1}^{V} (\mathbf{n_d})_j \log\left(\sum_{k=1}^{K} (\boldsymbol{\phi}_\mathbf{k})_j (\boldsymbol{\theta}_\mathbf{d})_k\right) + \sum_{d=1}^{M}\sum_{k=1}^{K} ((\boldsymbol{\alpha})_k - 1)\cdot\log((\boldsymbol{\theta}_\mathbf{d})_k) + \sum_{k=1}^{K}\sum_{j=1}^{V} \big((\boldsymbol{\beta})_j - 1\big)\cdot\log\big((\boldsymbol{\phi}_\mathbf{k})_j\big) \tag{8.5}$$

One can clearly see that as soon as any parameter estimate for $(\boldsymbol{\Theta}_\mathbf{d})_k$ for $d = 1, \dots, M$ and $k = 1, \dots, K$ or any $(\boldsymbol{\Phi}_\mathbf{k})_j$ for $k = 1, \dots, K$ and $j = 1, \dots, V$ equals zero, problems arise, since the log of that parameter will go to $-\infty$. If also either $(\boldsymbol{\alpha})_i$ or $(\boldsymbol{\beta})_i$ is smaller than one, the log posterior goes to $+\infty$. Indeed, that is the highest log posterior value and thus the posterior mode, but we cannot conclude anything about whether, in general, all parameters are estimated well or not.

This occurs several times in the estimations of VBEM, therefore we see in table 8.7 either +inf or -inf. Note that for $(\boldsymbol{\alpha})_i = \beta = 1$, the log posterior of VBEM gives back a ‘NaN’. That is caused by the fact that in the computation occurs a term $0 \cdot \infty$, which is undefined. Therefore, the log posterior measure does not help in making a good comparison between the results by VBEM, and by the other methods.

Considering the log posterior values of the optimization method and Gibbs sampling, we see that both exist. That is, all estimated parameters have a value larger than 0, albeit in the order of 10^{-30}. It is surprising that, although the explicit goal of Adam optimization is to find a maximum of the log posterior, the Gibbs sampling method finds parameters with a higher log posterior value in all cases in which either $(\boldsymbol{\alpha})_i$ or $(\boldsymbol{\beta})_i$ is smaller than one. From the previous example, it was already concluded that the optimization method does not always work properly if one of the hyperparameters is smaller than 1. This can also be concluded from table 8.7, where the log posterior value of KNIME is higher than the log posterior values of the optimization methods with both initializations for almost all settings in which either hyperparameter is smaller than 1. Only the settings $(\boldsymbol{\alpha})_i = 0.8$ and $(\boldsymbol{\beta})_i = 0.1$ and for $(\boldsymbol{\alpha})_i = 0.1$ and $(\boldsymbol{\beta})_i = 1$ are exceptions. With these settings, the optimization method with VBEM initialization performs best.

However, for both hyperparameters larger than 1, that is $(\boldsymbol{\alpha})_i = 1.1$ and $(\boldsymbol{\beta})_i = 1.1$, the optimization method with random initialization performs better, both in terms of log posterior value as in terms of perplexity. The optimization method with VBEM initialization, however, performs worse. The latter indicates that the starting point of the optimization algorithm can have a large influence on the model validation scores.

For this reason, the perplexity and log posterior scores are determined for the optimization method with different starting points. The random training and test set split is fixed using a seed and hyperparameters are chosen to be $(\boldsymbol{\alpha})_i = 0.999$ and $(\boldsymbol{\beta})_i = 0.999$, since for these settings, no truncation is applied. The results of this sensitivity test are given in table 8.8.

We see that there is a lot of variation in perplexity and log posterior values caused by a random starting point. However, none of the scores in table 8.8 can beat the performance of KNIME.

To check whether KNIME was not only lucky with its scores in table 8.7, a sensitivity test is performed by changing the seed in the Gibbs sampling algorithm. With different seeds, the Gibbs sampling algorithm draws different samples. The robustness of the algorithm can then be determined by computing the perplexity and log posterior values.

Although only 5 runs are done, we can already see that the variation in perplexity and log posterior scores is a lot smaller for KNIME than for the optimization method. Therefore, it is concluded that KNIME is a more robust algorithm than Adam optimization.

Another sensitivity test can be done by looking at the split of the documents in training and test sets. Note that in table 8.7, the same training and test sets were used for each method, and in each setting. This sensitivity test is only executed for the optimization method with random initialization. We see that the variation due to the random split in training and test set is smaller than the variation due to the random initialization and

Table 8.8: Perplexity and log posterior scores for LDA applied to 2000 reviews about strollers. The hyperparameters are set to $(\boldsymbol{\alpha})_i = 0.999$ and $(\boldsymbol{\beta})_i = 0.999$. There are ten topics to be found, thus $K = 10$. The Adam optimization algorithm is used with a stopping criterion threshold of 10^{-3} and random initialization. The same split in training and test set is used to avoid measuring multiple effects at the same time. The only variance comes from the initialization and the steps taken in the algorithm.

run	perplexity	log posterior
1	1160	-394845
2	1210	-401540
3	1154	-393955
4	1024	-386141
5	1157	-399764
6	948.7	-366133
7	927.0	-368490
8	1081	-387658
9	1165	-402124

Table 8.9: Perplexity and log posterior scores for LDA applied to 2000 reviews about strollers. The hyperparameters are set to $(\boldsymbol{\alpha})_i = 0.999$ and $(\boldsymbol{\beta})_i = 0.999$. There are ten topics to be found, thus $K = 10$. The KNIME *Topic Extractor* is used with different seeds. The training and test set are kept the same throughout the analysis.

run	perplexity	log posterior
1	688	-340473
2	684	-340481
3	687	-340441
4	688	-340379
5	674	-340465

the optimization algorithm itself. Again, comparing all the different outcomes from the sensitivity tests with KNIMEs results in table 8.7, KNIME keeps outperforming the optimization methods.

The remaining question is now, which method is preferred? One that gives back ‘flat’ probability vectors, such that the perplexity is the lowest and it has the highest predictive power (VBEM)? Or one in which the validness of the estimates relies on whether the algorithm does not work as it generally should, but in the end performs well for all hyperparameter settings (Gibbs)? Or one that finds the posterior mode the best, but only for hyperparameters larger than 1 (Adam optimization)?
The answer that follows from tests with both a small and large data set is that Gibbs sampling using KNIME works best. The method is robust, works for all hyperparameter settings, and is very quick due to smart parallel programming. However, the application in KNIME is challenging to adapt, as it is integrated into an interface within KNIME. Fortunately, the program is open source, so it is possible to dive into the code and make adjustments where desired. Nevertheless, for this master thesis, that is considered out of scope. The Gibbs sampling algorithm is given, and its performance is shown to be good, so if a fast application in a primary programming language like C is possible to construct, we recommend to use this method of inference. The optimization method does not perform poorly, especially not for hyperparameters larger than 1. Therefore, this algorithm is not ruled out. If there is no time nor possibility to do Gibbs sampling, Adam optimization using the package *Tensorflow* in Python and a fast GPU is well-suited, but the usage of different initializations is recommended, such that the best estimates in terms of the log posterior can be determined.

Interpretation of review results

To give an idea of what kind of conclusions can be drawn from LDA, a visualization of a topic is shown in figure 8.3. Similar figures for all other topics can be found in appendix B.1.

From the top 20 words of topic 5, we can already see that this topic is mainly about wheels, probably about the combination ‘front wheel’, and wheels that are locked. Furthermore, the most common word is ‘bob’, so the reviews are expected to be about the Bob stroller, with which you can jog. Also, both the word ‘turn’ and ‘revolution’ are frequently used. Although, these words can have different meanings, for the stroller reviews, we expect people to write about whether or not they can easily make a turn with the stroller.
Note that the word ‘stroller’ does not occur. This word is removed from the entire data set beforehand. Words

Table 8.10: Perplexity and log posterior scores for LDA applied to 2000 reviews about strollers. The hyperparameters are set to $(\boldsymbol{\alpha})_i = 0.99$ and $(\boldsymbol{\beta})_i = 0.99$. There are 10 topics to be found, thus $K = 10$. The Adam optimization algorithm is used with a threshold of 10^{-3} and random initialization. To avoid measuring multiple effects at the same time, the random initialization of Adam optimization is fixed using a seed. The training and test set division is now to only random effect, and it is checked that, for each run, the training and test sets are different.

run	perplexity	log posterior
1	964	-390348
2	1026	-400230
3	1178	-408713
4	1040	-400385
5	940	-393722
6	957	-390796
7	1067	-407518
8	890	-393323
9	1167	-407784
10	1072	-405006

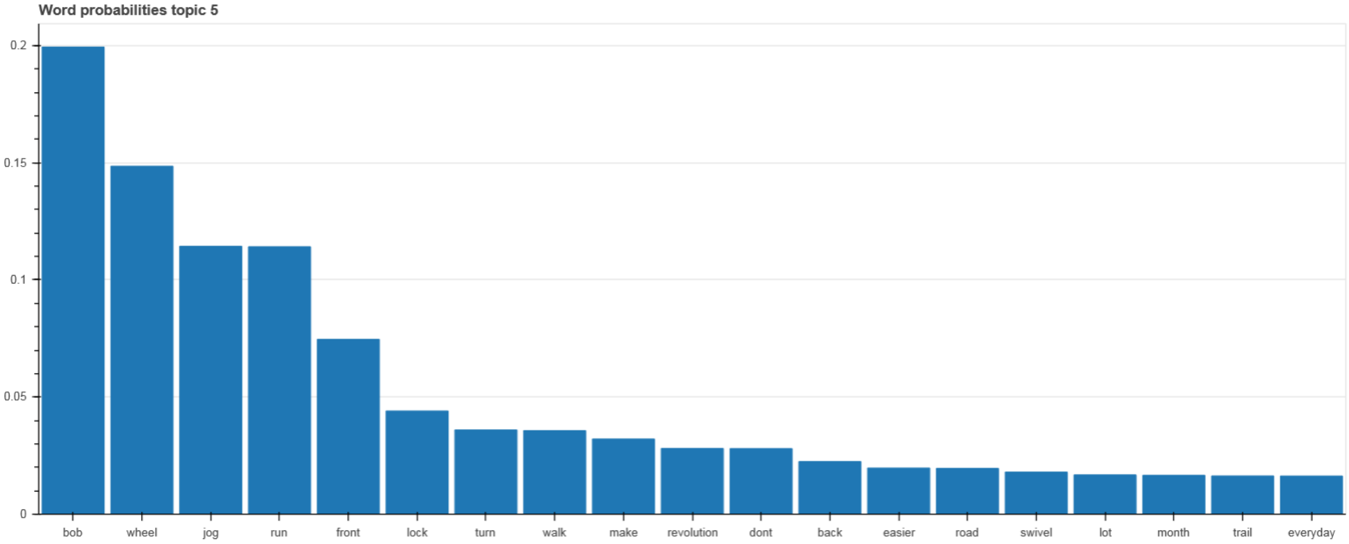


Figure 8.3: Word probabilities of the top 20 words of topic 5. This $\hat{\boldsymbol{\phi}}_5$ topic-word distribution is estimated using Adam optimization with $(\boldsymbol{\alpha})_i = 1.1$ and $(\boldsymbol{\beta})_i = 1.1$, and with random initialization.

that occur very often, and in this case in almost every review, can make the performance of LDA worse concerning topic interpretability. We already know that each document is about a stroller, so it does not give us more information when the word ‘stroller’ is included as top word in each topic-word distribution. Our interest is focused on what customers write about their stroller, what problems they encounter, and what they would like to see changed. Also, specific types or brand of strollers can stand out positively or negatively, which is something we would like to extract from the data. This phenomenon can, for example, be seen in topic 1, where among the top words are ‘city’, ‘jogger’, ‘mini’, ‘baby’, ‘britax’, ‘gt’, which form the names of a specific products. Topic 1 is therefore specifically about the ‘City Mini GT Jogger’ stroller and the ‘Britax’ car seat, as these are the names customers have used to refer to their strollers or car seats in their reviews.

When we want to know more about the stories behind topic 5 in figure 8.3, or we wish to gain further insights in the sentiment described, we can look at the top reviews that belong to that topic. That is, estimates for $\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_M$ are looked at, and the reviews that have the highest probability of belonging to topic 5 are selected. The top 6 reviews belonging mostly to topic 5 are given in appendix B.2. Note that the reviews can be long; two of them are not even given entirely because they are too long to visualize. It is an interesting fact that some customers tend to write a whole essay about the stroller, while others only say ‘*good*’. This variation needs to be taken into account before choosing the hyperparameters, and support the conclusion that $(\boldsymbol{\alpha})_i$ should be close to 1, such that a lot of variation between the number of topics per review is allowed for.

From the reviews, we can deduce that topic 5 is indeed about the Bob jogger stroller, with which you can easily jog. There are two modes of this stroller: walk and jog mode. In the jog mode, the front wheel is locked, such that the jogger keeps going straight. The customers think you can easily switch the front wheel from being locked to unlocked and are satisfied which this option. Also, they are happy with the front wheel being locked while jogging, as this makes jogging with it easier. Two customers write about the wrist strap that keeps the stroller attached to you when you are jogging. Both are not using it because they think it is dangerous to use. Only one customer is not satisfied, as his/her stroller's front wheel locks on its own every time, while this is not desired.

Instead of having to read 1000 reviews and manually summarize the major themes in the whole data set, LDA gives us the main topics. From the highest probability words (via estimates $\hat{\boldsymbol{\phi}}_{\mathbf{k}}$), the story of the topic can already be speculated about, but when reading the top reviews (determined from $\hat{\boldsymbol{\theta}}_{\mathbf{d}}$ estimates) for that specific topic, a more detailed story is retrieved. With this information, a next-generation type of strollers can be improved, or marketing strategies can be adjusted.

In chapter 7 about validity of the topic-word distribution estimates, it is said that even after having estimated model parameters $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ and getting interpretable results, attention needs to be paid to their validness. The number of topics K is chosen based on intuition and expectation. However, it is not certain that K fits the data. It might be the case that there are more topics hidden in the data than we have set in K, which results in two subjects being joined in one topic in the current model. On the other hand, when K is too large compared to the actual number of topics in the data set, some topics will fit noise. To this end, in chapter 7, the NKLS (normalized Kullback-Leibler divergence similarity) and NJSS (normalized symmetric Jensen-Shannon divergence similarity) measures are defined. These measures[2] indicate the extent of similarity between two topics. That is, if two topics both fit noise, they will be more similar than two topics that have their own stories. The NKLS scores are computed and summarized in table 8.11 for the estimates of all $\boldsymbol{\Phi}_{\mathbf{k}}$ for $k = 1, \ldots, K$ of the stroller data set determined with Adam optimization. The NJSS scores of the same estimates are shown in table 8.12.

Table 8.11: Normalized KL-divergence scores between all estimated topic-word distributions $\hat{\boldsymbol{\phi}}_{\mathbf{1}}, \ldots, \hat{\boldsymbol{\phi}}_{\mathbf{K}}$, which are determined for the stroller data using Adam optimization with symmetric $(\boldsymbol{\alpha})_i = 1.1$ and $(\boldsymbol{\beta})_i = 1.1$, and setting the number of topics $K = 10$.

	$\hat{\boldsymbol{\phi}}_0$	$\hat{\boldsymbol{\phi}}_1$	$\hat{\boldsymbol{\phi}}_2$	$\hat{\boldsymbol{\phi}}_3$	$\hat{\boldsymbol{\phi}}_4$	$\hat{\boldsymbol{\phi}}_5$	$\hat{\boldsymbol{\phi}}_6$	$\hat{\boldsymbol{\phi}}_7$	$\hat{\boldsymbol{\phi}}_8$	$\hat{\boldsymbol{\phi}}_9$
$\hat{\boldsymbol{\phi}}_0$	1	0.262	0.105	0.114	0.584	0.334	0.378	0.483	0.327	0.465
$\hat{\boldsymbol{\phi}}_1$	0.262	1	0.049	0.106	0.354	0.324	0.322	0.395	0.193	0.421
$\hat{\boldsymbol{\phi}}_2$	0.105	0.049	1	0	0.359	0.069	0.281	0.132	0.104	0.146
$\hat{\boldsymbol{\phi}}_3$	0.113	0.106	0	1	0.185	0.073	0.211	0.190	0.010	0.197
$\hat{\boldsymbol{\phi}}_4$	0.584	0.354	0.359	0.185	1	0.446	0.550	0.580	0.456	0.565
$\hat{\boldsymbol{\phi}}_5$	0.334	0.324	0.069	0.073	0.446	1	0.424	0.452	0.241	0.327
$\hat{\boldsymbol{\phi}}_6$	0.377	0.322	0.281	0.211	0.550	0.424	1	0.463	0.392	0.349
$\hat{\boldsymbol{\phi}}_7$	0.483	0.395	0.132	0.190	0.580	0.452	0.463	1	0.459	0.639
$\hat{\boldsymbol{\phi}}_8$	0.327	0.193	0.104	0.010	0.456	0.241	0.391	0.459	1	0.454
$\hat{\boldsymbol{\phi}}_9$	0.465	0.421	0.146	0.197	0.565	0.327	0.349	0.639	0.455	1

There are no remarkable differences between the NKLS and NJSS scores in tables 8.11 and 8.12. All non-zero and non-one NJSS scores are smaller than the corresponding NKLS scores, but, in general, the same conclusions can be drawn from both tables. Therefore, we will only focus on the NKLS scores from table 8.11. Naturally, there are diagonals with NKLS = 1, as each topic-word distribution is exactly the same as itself. Furthermore, there are two zeros for the combination $\hat{\boldsymbol{\phi}}_{\mathbf{2}}$ and $\hat{\boldsymbol{\phi}}_{\mathbf{3}}$, meaning that these two topic-word distributions are the most different among all combinations. All other NKLS values are in between, meaning that the closer to zero, the more similar two topic-word probability vectors are. In chapter 7, we have said that when a NKLS score is higher than 0.9, two $\hat{\boldsymbol{\phi}}$s are considered to be about the same topic. Fortunately, in table 8.11 all similarity scores (except the diagonal of course) are below 0.9, meaning that each topic can be interpreted separately. This might even be an indicator that K can be increased, as we are not fitting noise yet.

It is wise to always perform this check after having estimated all topic-word distributions. In this way, one can draw better conclusions, and noise is observed beforehand, instead of after having read the top reviews for

[2] They are not measures in the mathematical sense.

Table 8.12: Normalized symmetric JS-divergence similarity scores between all estimated topic-word distributions $\hat{\boldsymbol{\phi}}_\mathbf{1},\ldots,\hat{\boldsymbol{\phi}}_\mathbf{K}$, which are determined for the stroller data using Adam optimization with symmetric $(\boldsymbol{\alpha})_i = 1.1$ and $(\boldsymbol{\beta})_i = 1.1$, and setting the number of topics $K = 10$.

	$\hat{\boldsymbol{\phi}}_\mathbf{0}$	$\hat{\boldsymbol{\phi}}_\mathbf{1}$	$\hat{\boldsymbol{\phi}}_\mathbf{2}$	$\hat{\boldsymbol{\phi}}_\mathbf{3}$	$\hat{\boldsymbol{\phi}}_\mathbf{4}$	$\hat{\boldsymbol{\phi}}_\mathbf{5}$	$\hat{\boldsymbol{\phi}}_\mathbf{6}$	$\hat{\boldsymbol{\phi}}_\mathbf{7}$	$\hat{\boldsymbol{\phi}}_\mathbf{8}$	$\hat{\boldsymbol{\phi}}_\mathbf{9}$
$\hat{\boldsymbol{\phi}}_\mathbf{0}$	1	0.253	0.087	0.106	0.494	0.320	0.323	0.451	0.332	0.458
$\hat{\boldsymbol{\phi}}_\mathbf{1}$	0.252	1	0.034	0.130	0.336	0.347	0.267	0.407	0.237	0.415
$\hat{\boldsymbol{\phi}}_\mathbf{2}$	0.087	0.034	1	0	0.283	0.043	0.211	0.148	0.073	0.128
$\hat{\boldsymbol{\phi}}_\mathbf{3}$	0.106	0.130	0	1	0.176	0.103	0.211	0.195	0.015	0.238
$\hat{\boldsymbol{\phi}}_\mathbf{4}$	0.493	0.336	0.283	0.176	1	0.453	0.498	0.577	0.449	0.532
$\hat{\boldsymbol{\phi}}_\mathbf{5}$	0.320	0.347	0.043	0.103	0.453	1	0.414	0.457	0.299	0.365
$\hat{\boldsymbol{\phi}}_\mathbf{6}$	0.323	0.267	0.211	0.211	0.498	0.414	1	0.456	0.361	0.338
$\hat{\boldsymbol{\phi}}_\mathbf{7}$	0.451	0.407	0.148	0.195	0.577	0.457	0.456	1	0.460	0.619
$\hat{\boldsymbol{\phi}}_\mathbf{8}$	0.332	0.237	0.073	0.015	0.449	0.299	0.361	0.460	1	0.441
$\hat{\boldsymbol{\phi}}_\mathbf{9}$	0.457	0.415	0.128	0.238	0.532	0.365	0.338	0.619	0.441	1

each topic, and not being able to retrieve a coherent story behind the topic. As the difference between the NKLS and NJSS scores are so small, the similarity score based on the symmetrized Kullback-Leibler divergence is preferred, as a more thorough study is performed in [24].

8.3. LDA with syntax and sentiment: is this the future?

The model of LDA with syntax and sentiment is an extension of basic LDA; thus, the parameter space in which we search for the posterior mode location is larger. In this extension, we look for a topic distribution per document, a sentiment distribution per document, and word distributions per topic-sentiment combination. In LDA with syntax and sentiment, a topic is drawn per phrase instead of per word as in plain LDA. Also, a sentiment is assigned to each phrase, and all words in a phrase or sentence are drawn from the word distribution for the corresponding topic-sentiment combination belonging to that phrase. Although different methods of inference were researched for basic LDA, in this extension, only Adam optimization is used to find the posterior mode estimates for respectively $\mathbf{\Theta_d}$ for $d = 1,\dots,M$, $\mathbf{\Pi_d}$ for $d = 1,\dots,M$, and $\mathbf{\Phi_{k,o}}$ for $k = 1,\dots,K$ and $o = 1,\dots,\Sigma$.

The increased number of parameters to be estimated and the addition of sentiments make the form of the posterior density for this LDA extension more complicated compared to the one for basic LDA. This results in a slower convergence in the optimization. Furthermore, a count array with the frequencies of each word per phrase and per document needs to be computed. The latter is the most significant bottleneck encountered in the inference of LDA with syntax and sentiment. This count matrix[3] has proportions that large, that the used computer server cannot handle it in terms of memory. The count matrix consists of floating points with 32 bits, which is required for steps in the optimization algorithm to prevent accuracy loss. For the application of LDA with syntax and sentiment to 200 reviews with at most 100 phrases (otherwise they are removed from the data set), and a vocabulary size of 1000, the count matrix has $200 \cdot 100 \cdot 1000 = 2 \cdot 10^7$ elements, and each element is a floating point of 32 bits. This results in a count matrix that is too large to keep in memory. Although the array consists of many zeros, it is not possible to convert it to a sparse array because the Adam optimization implementation in *Tensorflow* cannot handle sparse arrays. For this reason, LDA with syntax and sentiment in the current implementation can only be used for small data sets.

In this section, first, the algorithm and model are tested for a simulated data set of which the model parameters are known. Then, we gain more intuition of the model by its application to a toy data set. Consequently, with more knowledge about which settings to use in Adam optimization to obtain good estimates, the algorithm is tested on the stroller data set. Unfortunately, only a minimal number of reviews can be used due to memory problems.

8.3.1. Model testing on gibberish

First, we generate a small data set following the generative process of LDA with syntax and sentiment from chapter 6. A data set has been created consisting of 20 documents with 2 hidden topics. As usual, there are three sentiments: positive, neutral and negative. There are 26 words in the vocabulary, from which 5 are positive, 9 are negative, and the rest is neutral. Because we want distinct topics that can relatively easily be found by the inference method, hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ are symmetric and set to respectively $(0.5, 0.5)$ and $(0.5, 0.5, 0.5)$. These form the parameters of a Dirichlet prior. With all parameters being smaller than 1, we expect the documents to be mostly about a single topic and to have one sentiment. The last hyperparameter is $\boldsymbol{\beta_o}$ for $o = 1,2,3$. In $\boldsymbol{\beta_1}$, the hyperparameter vector for the positive sentiment topic-word distribution, 50% of the probability mass is given to the positive words, and the other half is given to the neutral and negative words together. Then, in $\boldsymbol{\beta_2}$, 70% of the probability mass is given to the neutral words, and 30% to the positive and negative words. A higher percentage is chosen here because there are more neutral words in the vocabulary than positive words, and we want the values of each $\boldsymbol{\beta_o}$ to be of the same order of magnitude. Lastly, in $\boldsymbol{\beta_3}$, 60% of the probability mass is given to the negative words, and the rest to the positive and neutral words.
With these hyperparameters, we can draw $\mathbf{\Theta_d}$ from Dirichlet($\boldsymbol{\alpha}$) and $\mathbf{\Pi_d}$ from Dirichlet($\boldsymbol{\gamma}$) for each document $d \in \{1,\dots,M\}$. Subsequently, for each $o \in \{1,2,3\}$ and $k = 1,2$, we draw $\mathbf{\Phi_{k,o}}$ from Dirichlet($\boldsymbol{\beta_o}$). To construct documents, we first need to draw a number of phrases S_d for each document. This is done using a Poisson(2) distribution. Each document consists of at least 10 sentences plus S, where $S \sim$ Poisson(2). Also the number of words per phrase is random, and is drawn from a Poisson(1) distribution. The minimal number of words is 5, to which the Poisson draw is added.
Consequently, with all $\boldsymbol{\theta_d}$, $\boldsymbol{\pi_d}$ and S_d being generated, we can draw a topic and a sentiment for each phrase in each document. Then, given the topic-sentiment combination and the number of words in each sentence, words can be drawn. This results in a gibberish data set as can be seen in tables B.2 and B.3 in appendix B.3.

[3] Strictly speaking, this is an array with size $N \times \max_d\{S_d\} \times V$. In *Tensorflow*, the variable type is a tensor with the aforementioned shape.

However, the values of the latent random variables in the model, $\mathbf{\Theta_d}$ for $d = 1,\ldots,M$, $\mathbf{\Pi_d}$ for $d = 1,\ldots,M$, and $\mathbf{\Phi_{k,o}}$ for $k = 1,\ldots,K$ and $o = 1,\ldots,\Sigma$, are known, and we can check if the posterior mode estimates determined by Adam optimization correspond to the true values.

Adam optimization for LDA with syntax and sentiment has to deal with a large parameter space in which it searches for find the maximum (i.e. the posterior mode). If we set the values of one hyperparameter vector $\boldsymbol{\alpha}$ or $\boldsymbol{\gamma}$ smaller than 1, it quickly falls into the abyss at the boundaries, as explained in the previous section for Adam optimization applied to basic LDA. Therefore, we choose the values of $(\boldsymbol{\alpha})_i$ and $(\boldsymbol{\gamma})_i$ to be larger than 1. The best posterior mode estimates for $\mathbf{\Theta}$ and $\mathbf{\Pi}$ are obtained for $\boldsymbol{\alpha} = (1.8, 1.8)$ and $\boldsymbol{\gamma} = (1.1, 1.1)$. From runs with different hyperparameter choices, we conclude that the algorithm quickly 'decides' that a document belongs to only one topic, even if we set $(\boldsymbol{\alpha})_i$ larger than 1. Therefore, this hyperparameter vector has larger values than $\boldsymbol{\gamma}$. It is more challenging to find the settings of hyperparameter vectors $\boldsymbol{\beta_o}$ with $o = 1,2,3$ for which the posterior mode estimates correspond with the true parameters. Empirically, we found that the best settings for $\boldsymbol{\beta_o}$ with $o = 1,2,3$ are obtained if they are constructed by applying 90% of the probability mass to respectively the positive, neutral, or negative words. Consequently, the obtained vectors are multiplied by 50 for the positive hyperparameter vector $\boldsymbol{\beta_1}$, and 40 for both the neutral and negative hyperparameter vectors $\boldsymbol{\beta_2}$ and $\boldsymbol{\beta_3}$.

But how can we determine which estimates are the best? For the computation of the perplexity, we need to split up the data set into a training and test set. Because LDA with syntax and sentiment focuses on topics and sentiments on a sentence-level, and, in practice, many reviews consists only of a few sentences, it is not wise to split. Therefore, we have reported the estimates of $\mathbf{\Theta_d}$ for $d = 1,\ldots,M$, $\mathbf{\Pi_d}$ for $d = 1,\ldots,M$, and $\mathbf{\Phi_{k,o}}$ for $k = 1,\ldots,K$ and $o = 1,\ldots,\Sigma$, and compared them manually with the corresponding true parameters.

Firstly, we take a look at the posterior mode estimates of all document-topic distributions. In table 8.13, the true values of $\boldsymbol{\theta_d}$ for $d = 1,\ldots,M$ are shown, and in table 8.14, the corresponding estimates are given.

Table 8.13: Simulated document-topic distributions that are independently drawn from a Dirichlet(0.5,0.5) distribution.

d	$(\boldsymbol{\theta_d})_1$	$(\boldsymbol{\theta_d})_2$
1	0.176	0.824
2	0.878	0.122
3	0.999	0.001
4	0.971	0.029
5	0.332	0.668
6	0.821	0.179
7	0.839	0.161
8	0.989	0.011
9	0.973	0.027
10	0.973	0.027
11	0.017	0.983
12	0.244	0.756
13	0.599	0.401
14	0.979	0.021
15	0.013	0.987
16	0.931	0.069
17	0.781	0.219
18	1.000	0.000
19	0.426	0.574
20	0.932	0.068

Table 8.14: Posterior mode estimates of the document-topic distributions of the simulated data, determined with Adam optimization in which the hyperparameter settings are $(\boldsymbol{\alpha})_i = 1.8$, $(\boldsymbol{\gamma})_i = 1.1$, and $\boldsymbol{\beta_o}$ for $o = 1,2,3$ constructed as described above. A learning rate of 0.0001 and random initialization have been used.

d	$(\hat{\boldsymbol{\theta}}_\mathbf{d})_1$	$(\hat{\boldsymbol{\theta}}_\mathbf{d})_2$
1	0.059	0.941
2	0.931	0.069
3	0.941	0.059
4	0.945	0.055
5	0.245	0.755
6	0.818	0.182
7	0.845	0.155
8	0.931	0.069
9	0.945	0.055
10	0.778	0.222
11	0.059	0.941
12	0.534	0.466
13	0.651	0.349
14	0.941	0.059
15	0.055	0.945
16	0.868	0.132
17	0.759	0.241
18	0.931	0.069
19	0.222	0.778
20	0.857	0.143

The estimates are considered good if the following principles are satisfied. The allocation of the main topic corresponds to the true allocation, that is, if a document is mainly about one topic (for example document 3), this is also the case in the estimation results. Secondly, if a document is about both topics (for example topic

13), the same phenomenon can be seen in the estimates, albeit in a slightly different proportion.
With these two weak principles, we can conclude that the estimates of $\boldsymbol{\Theta}_\mathbf{d}$ by Adam optimization perform relatively well. The main topic allocations, and documents that contain both topics are found.

Subsequently, the estimates of the document-sentiment distributions $\hat{\boldsymbol{\pi}}$ are compared with the corresponding true sentiment distributions in tables 8.15 and 8.16. Similarly as for the document-topic distributions, we check if the main sentiments are found in the posterior mode estimates. We see in table 8.16 that if a document has almost only one sentiment, this effect is also given in the estimates. If there are multiple sentiments, the $\hat{\boldsymbol{\pi}}_\mathbf{d}$, deviate a bit more from the true $\boldsymbol{\pi}_\mathbf{d}$ values. In general, we can conclude that the main effects are captured by the Adam optimization.

Table 8.15: Simulated document-sentiment distributions that are independently drawn from a Dirichlet(0.5, 0.5, 0.5) distribution.

d	$(\hat{\boldsymbol{\pi}}_\mathbf{d})_1$	$(\hat{\boldsymbol{\pi}}_\mathbf{d})_2$	$(\hat{\boldsymbol{\pi}}_\mathbf{d})_3$
1	0.294	0.699	0.006
2	0.414	0.435	0.150
3	0.006	0.010	0.984
4	0.891	0.083	0.026
5	0.031	0.810	0.159
6	0.267	0.460	0.273
7	0.087	0.045	0.868
8	0.000	0.924	0.076
9	0.862	0.092	0.045
10	0.387	0.609	0.004
11	0.775	0.009	0.216
12	0.091	0.218	0.691
13	0.072	0.913	0.015
14	0.208	0.357	0.435
15	0.051	0.920	0.029
16	0.776	0.217	0.007
17	0.693	0.047	0.260
18	0.977	0.016	0.007
19	0.000	0.396	0.604
20	0.516	0.091	0.393

Table 8.16: Posterior mode estimates of the document-sentiment distributions of the simulated data, determined with Adam optimization in which the hyperparameter settings are $(\boldsymbol{\alpha})_i = 1.8$, $(\boldsymbol{\gamma})_i = 1.1$, and $\boldsymbol{\beta}_\mathbf{o}$ for $o = 1, 2, 3$ constructed as described above. A learning rate of 0.0001 and random initialization have been used.

d	$(\hat{\boldsymbol{\pi}}_\mathbf{d})_1$	$(\hat{\boldsymbol{\pi}}_\mathbf{d})_2$	$(\hat{\boldsymbol{\pi}}_\mathbf{d})_3$
1	0.378	0.614	0.008
2	0.495	0.301	0.204
3	0.008	0.008	0.984
4	0.834	0.158	0.008
5	0.073	0.778	0.149
6	0.383	0.383	0.233
7	0.107	0.013	0.880
8	0.010	0.883	0.107
9	0.979	0.014	0.008
10	0.274	0.717	0.009
11	0.975	0.009	0.016
12	0.083	0.233	0.684
13	0.007	0.987	0.007
14	0.415	0.252	0.333
15	0.008	0.975	0.017
16	0.901	0.091	0.008
17	0.495	0.010	0.495
18	0.949	0.042	0.010
19	0.009	0.452	0.540
20	0.097	0.009	0.894

Lastly, the topic-sentiment-word distributions are estimated using the posterior mode determination via Adam optimization. In tables 8.17 and 8.18, the true values of $\boldsymbol{\Phi}$ used to generate the data, and the posterior mode estimates are given.
In this case, estimates are considered good if the main words per $\hat{\boldsymbol{\phi}}_\mathbf{k,o}$ for each possible (k, o)-combination are the same, and if the word probabilities are of the same order for the top 10 words in each distribution. As an extra check, the NJSS similarity scores between all true $\boldsymbol{\phi}_\mathbf{k,o}$ and estimated $\hat{\boldsymbol{\phi}}_\mathbf{k,o}$ are given in table 8.19. From both manual comparison of the true topic-sentiment-word distributions with the estimates and NJSS similarity scores, we see that the estimates are very similar to the true $\boldsymbol{\phi}$ distributions. Therefore, we conclude that Adam optimization used for posterior mode determination for LDA with syntax and sentiment works well for simulated data.

Table 8.17: Simulated topic-sentiment-word distributions that are independently drawn from Dirichlet($\boldsymbol{\beta_0}$) distributions corresponding to their sentiment. Hyperparameters $\boldsymbol{\beta_0}$ are constructed by dividing the probability mass over the corresponding sentiment words and the rest of the vocabulary.

Vocabulary	$\hat{\boldsymbol{\phi}}_{1,1}$	$\hat{\boldsymbol{\phi}}_{1,2}$	$\hat{\boldsymbol{\phi}}_{1,3}$	$\hat{\boldsymbol{\phi}}_{2,1}$	$\hat{\boldsymbol{\phi}}_{2,2}$	$\hat{\boldsymbol{\phi}}_{2,3}$
afraid	0.000	0.052	0.310	0.004	0.000	0.001
aggressive	0.000	0.000	0.003	0.000	0.000	0.004
allergic	0.000	0.008	0.000	0.000	0.000	0.000
canid	0.198	0.069	0.000	0.000	0.000	0.068
cat	0.028	0.532	0.000	0.000	0.009	0.000
dog	0.002	0.000	0.000	0.001	0.000	0.000
donot	0.000	0.013	0.010	0.522	0.000	0.000
fluffy	0.000	0.002	0.000	0.000	0.030	0.000
give	0.000	0.000	0.000	0.000	0.509	0.135
happiness	0.034	0.000	0.000	0.141	0.000	0.000
hate	0.084	0.015	0.000	0.001	0.000	0.000
lifelong	0.264	0.000	0.000	0.000	0.000	0.000
like	0.000	0.003	0.000	0.009	0.000	0.028
make	0.000	0.025	0.000	0.000	0.000	0.067
most	0.145	0.000	0.000	0.000	0.000	0.000
nice	0.001	0.000	0.000	0.220	0.000	0.046
noteworthy	0.143	0.001	0.000	0.001	0.000	0.000
people	0.000	0.005	0.077	0.098	0.201	0.000
pet	0.077	0.202	0.000	0.000	0.000	0.271
regret	0.000	0.000	0.001	0.000	0.000	0.004
sad	0.000	0.000	0.239	0.000	0.000	0.321
smell	0.000	0.022	0.359	0.002	0.000	0.053
stubborn	0.020	0.000	0.000	0.000	0.000	0.000
stupid	0.000	0.000	0.000	0.000	0.000	0.000
walk	0.000	0.008	0.000	0.000	0.247	0.002
wet	0.004	0.041	0.000	0.000	0.003	0.001

Table 8.18: Posterior mode estimates of the topic-sentiment-word distributions of the simulated data, determined with Adam optimization in which the hyperparameter settings are $(\boldsymbol{\alpha})_i = 1.8$, $(\boldsymbol{\gamma})_i = 1.1$, and $\boldsymbol{\beta_o}$ for $o = 1,2,3$ constructed as described above. A learning rate of 0.0001 and random initialization have been used.

Vocabulary	$\hat{\boldsymbol{\phi}}_{1,1}$	$\hat{\boldsymbol{\phi}}_{1,2}$	$\hat{\boldsymbol{\phi}}_{1,3}$	$\hat{\boldsymbol{\phi}}_{2,1}$	$\hat{\boldsymbol{\phi}}_{2,2}$	$\hat{\boldsymbol{\phi}}_{2,3}$
afraid	0.004	0.041	0.291	0.011	0.007	0.015
aggressive	0.004	0.006	0.006	0.011	0.007	0.024
allergic	0.004	0.016	0.006	0.011	0.007	0.016
canid	0.173	0.065	0.001	0.003	0.002	0.019
cat	0.015	0.443	0.001	0.003	0.005	0.004
dog	0.005	0.002	0.001	0.003	0.002	0.004
donot	0.001	0.020	0.010	0.336	0.002	0.004
fluffy	0.001	0.002	0.001	0.003	0.032	0.004
give	0.001	0.002	0.002	0.003	0.424	0.096
happiness	0.049	0.021	0.018	0.152	0.022	0.051
hate	0.093	0.016	0.006	0.011	0.007	0.016
lifelong	0.249	0.021	0.018	0.035	0.022	0.051
like	0.013	0.030	0.018	0.035	0.022	0.066
make	0.001	0.020	0.001	0.003	0.002	0.035
most	0.146	0.002	0.001	0.003	0.002	0.004
nice	0.013	0.021	0.018	0.234	0.022	0.082
noteworthy	0.123	0.021	0.018	0.035	0.022	0.051
people	0.001	0.011	0.068	0.056	0.159	0.004
pet	0.063	0.153	0.001	0.003	0.002	0.162
regret	0.004	0.006	0.006	0.011	0.007	0.024
sad	0.004	0.006	0.197	0.011	0.007	0.166
smell	0.004	0.029	0.299	0.016	0.007	0.079
stubborn	0.021	0.006	0.006	0.011	0.007	0.016
stupid	0.000	0.000	0.000	0.000	0.000	0.000
walk	0.001	0.005	0.001	0.003	0.203	0.004
wet	0.003	0.036	0.001	0.003	0.002	0.004

Table 8.19: NJSS for columns in estimated and row simulated

NJSS	$\hat{\boldsymbol{\phi}}_{1,1}$	$\hat{\boldsymbol{\phi}}_{1,2}$	$\hat{\boldsymbol{\phi}}_{1,3}$	$\hat{\boldsymbol{\phi}}_{2,1}$	$\hat{\boldsymbol{\phi}}_{2,2}$	$\hat{\boldsymbol{\phi}}_{2,3}$
$\boldsymbol{\phi}_{1,1}$	0.996	0.652	0.975	0.978	0.526	0.879
$\boldsymbol{\phi}_{1,2}$	0.514	0.992	0.433	0.442	0.000	0.349
$\boldsymbol{\phi}_{1,3}$	0.982	0.517	0.997	0.994	0.534	0.892
$\boldsymbol{\phi}_{2,1}$	0.982	0.517	0.997	0.994	0.534	0.892
$\boldsymbol{\phi}_{2,2}$	0.456	0.040	0.458	0.467	0.993	0.756
$\boldsymbol{\phi}_{2,3}$	0.845	0.384	0.865	0.868	0.871	0.991

8.3.2. Stroller data set: deep dive on a single topic

The implementation of the inference method used to determine the model parameters of LDA with syntax and sentiment, Adam optimization, does not work well with large data sets, as explained in the beginning of this section. Due to memory problems, we need to restrict ourselves to analyses with at most 400 documents. Moreover, documents with more than 100 phrases are removed, as they blow up the size of the count array in which the observed words are summarized. Taking 400 documents out of the entire data set will result in inaccurate results because these documents will be about at least 10 topics, as shown in the previous section in which LDA is applied to the entire stroller data set. The estimation of 10 topics with each 3 possible sentiments out of a data set with 400 documents is infeasible. Therefore, we have chosen to pick a subset of the whole data set based on the topic the documents belong to.

Each document has an estimated probability vector $\hat{\boldsymbol{\theta}}_\mathbf{d}$ with the topic probabilities. For consistency with the previous results of plain LDA, we have selected the reviews that have most probability assigned to topic 5, that is $\arg\max_k(\hat{\boldsymbol{\theta}}_\mathbf{d})_k = 5$. This reduced stroller data set consists of 177 documents. We already know that together, these documents form a topic about the Bob Jogger stroller that has a front wheel that can be locked in order to facilitate jogging. Also, people write about the wrist strap. Therefore, we expect that there are 2 subtopics within this dataset, so $K = 2$. Again, there are three sentiments: $\Sigma = 3$. Using the previous results of LDA with syntax and sentiment with Adam optimization as inference method, we have set the hyperparameters $(\boldsymbol{\alpha})_i$ and $\boldsymbol{\gamma}$ both to 2. The probability mass in each vector $\boldsymbol{\beta}$ is divided 95%-5% for respectively the corresponding sentiment words and the remaining words in the vocabulary. Subsequently, $\boldsymbol{\beta}$ is multiplied by 10. The resulting estimates for $\boldsymbol{\Phi}_\mathbf{k,o}$ for topics $k = 1,2$ and sentiments $o = 1,2,3$ are given in tables 8.20 and 8.21. Note that the vectors $\boldsymbol{\phi}_\mathbf{k,o}$ are rearranged to display only the top 20 words with the highest probability in each word distribution.

Table 8.20: Estimated topic-sentiment-word distributions for subtopic 1 for stroller reviews that belong to topic 5 of an analysis with plain LDA (with hyperparameters $(\boldsymbol{\alpha})_i = 1.1$, $(\boldsymbol{\beta})_i = 1.1$). In this analysis for LDA with syntax and sentiment, $(\boldsymbol{\alpha})_i = 2$, $(\boldsymbol{\gamma})_i = 2$, and $\boldsymbol{\beta}$ is constructed as explained in section 8.3.2.

positive	$\hat{\boldsymbol{\phi}}_{\mathbf{1,1}}$	neutral	$\hat{\boldsymbol{\phi}}_{\mathbf{1,2}}$	negative	$\hat{\boldsymbol{\phi}}_{\mathbf{1,3}}$
love	0.024	bob	0.043	break	0.022
glad	0.022	wheel	0.038	pain	0.019
easy	0.021	run	0.033	problem	0.018
perfect	0.021	jog	0.024	useless	0.018
fine	0.020	front	0.023	pricey	0.018
sturdy	0.020	baby	0.022	hurt	0.018
smooth	0.018	make	0.018	worse	0.018
great	0.016	lock	0.016	cons	0.018
pretty	0.016	buy	0.015	downside	0.018
awesome	0.016	love	0.015	shock	0.017
happy	0.016	walk	0.014	mind	0.017
pleased	0.015	like	0.013	warning	0.017
hope	0.015	dont	0.012	slow	0.017
kid	0.015	great	0.012	tire	0.016
agree	0.015	jogger	0.012	frustrate	0.016
deal	0.015	im	0.011	long	0.016
decent	0.014	fold	0.011	barely	0.016
solid	0.014	seat	0.010	regret	0.016
drive	0.013	easy	0.010	suck	0.016
active	0.013	month	0.010	narrower	0.016
good	0.013	turn	0.010	disappointed	0.015

Table 8.21: Topic-sentiment-word distributions for subtopic 2 for stroller reviews that belong to topic 5 of an analysis with plain LDA (with hyperparameters $(\boldsymbol{\alpha})_i = 1.1$, $(\boldsymbol{\beta})_i = 1.1$). In the analysis for LDA with syntax and sentiment, $(\boldsymbol{\alpha})_i = 2$, $\gamma = 2$, and $\boldsymbol{\beta}$ is constructed as explained in section 8.3.2.

positive	$\hat{\boldsymbol{\phi}}_{\mathbf{2,1}}$	neutral	$\hat{\boldsymbol{\phi}}_{\mathbf{2,2}}$	negative	$\hat{\boldsymbol{\phi}}_{\mathbf{2,3}}$
versatile	0.014	bring	0.011	bulky	0.018
live	0.013	sister	0.010	disappointed	0.017
solid	0.013	travel	0.009	hang	0.016
simple	0.013	bigger	0.009	rough	0.015
fantastic	0.012	sand	0.009	issue	0.015
impress	0.012	toy	0.009	hole	0.015
favorite	0.012	return	0.009	warning	0.015
handy	0.012	sell	0.009	expensive	0.014
strong	0.011	wife	0.009	cheap	0.014
stable	0.011	rock	0.009	worry	0.013
real	0.010	house	0.009	buckle	0.013
helpful	0.010	product	0.008	worn	0.013
suggest	0.010	cost	0.008	miss	0.013
excite	0.010	anymore	0.008	lie	0.013
special	0.010	speed	0.008	backward	0.013
appreciate	0.010	wet	0.008	dirt	0.013
beautiful	0.010	room	0.008	rack	0.013
plenty	0.010	warn	0.008	cold	0.013
beautifully	0.010	aisle	0.008	narrow	0.013
importantly	0.010	order	0.008	disappoint	0.013
clear	0.010	part	0.008	flimsy	0.013

From tables 8.20 and 8.21, we can conclude that the positive, neutral and negative words are well divided over the three classes. This is forced by the assignment of much more probability mass to the words corresponding to the sentiment class than to the other words in the construction of hyperparameter vectors $\boldsymbol{\beta}_{\mathbf{o}}$. Furthermore, the two topics are found within this small stroller data set are distinctive, because they do not have the same top 20 words for either word distribution. A remark needs to be made here because we have to be careful interpreting these results, since the word probabilities are almost equal in each vector. We expect different results for other values for hyperparameters $\boldsymbol{\beta}$.
Focusing on table 8.20, we can conclude already that people are glad, happy, and pleased with their Bob jogger; find it pretty and sturdy, that they actually say they love it. On the downside, there is a problem with pain. Something apparently breaks and people mention that they are hurt. From the word list, it does not become immediately clear what actually hurts or break, so for more detailed knowledge, we would still have to return to reading the reviews. At the moment, we can, however, conclude more based on these tables, than on the word probabilities from plain LDA in the previous section.

In chapter 6, LDA with syntax and sentiment was proposed with a split on parts-of-speech. This can be done with the results in tables 8.20 and 8.21 by splitting each topic-sentiment-word distribution into 2 groups: one group with adjectives and adverbs, and one group with nouns and verbs. In this way, globally, we will get a list with opinion words (group 1) and a list with aspect words (group 2) for each topic-sentiment-word distribution $\hat{\boldsymbol{\phi}}_{\mathbf{k,o}}$. However, this split is not applied in this example because the estimates are considered too unreliable.

The test of inference with Adam optimization for the extended LDA model on a small stroller data set shows that there is potential. However, more research needs to be done on which settings are optimal, and on a measure that quantitatively determines whether one set of settings gives better results than another. Note that the perplexity score can again be used here, but the division of the data in a training and test set is too cumbersome for this model, as we need to take the sentences into account. Also, in the previous section, the perplexity has shown to have undesired properties. Therefore, it is not applied to the results for this LDA extension.

9 Discussion

"I have not failed. I've just found 10,000 ways that won't work."
Thomas Edison (1847-1931)

In this master thesis, topic model Latent Dirichlet Allocation is thoroughly researched, and different methods of inference are applied to estimate the desired model parameters. Throughout the process, several questions have arisen, and the LDA results for two data sets using different methods of inference have shown that some of them performs not satisfactorily, caused by several phenomena on which will be elaborated in the first section of this chapter. Shortcomings and arisen questions of the extension of LDA, LDA with syntax and sentiment, are explained and described in section 9.2.

9.1. Latent Dirichlet Allocation

9.1.1. Assumptions

First of all, the assumptions made in LDA are quite strong. Each review is considered to consist of a list of words that have no syntactic relation with each other, which means that all words can be permuted, while it is assumed that no information is lost. This bag-of-words assumption is the weakest part of LDA because these permutations result in weak topics. Naturally, some words are strictly linked to one topic, but many of them only have a clear meaning in relation to the words surrounding them in a sentence. Words like 'the' and 'for' are removed beforehand because they are considered to be 'stop words', but other frequently occurring words that are, for example, adjectives are still included in the data set. These words can have different meanings depending on the context, which is entirely lost in the bag-of-words representation.
To improve upon this aspect of LDA, the extension with syntax and sentiment is invented. The bag-of-words representation is still used, but now only on sentence level. However, the exchangeability of words in only applicable to the sentence or phrase level. With this splitting of documents into phrases, less information is lost, and the meaning of adjectives is linked to words in that same phrase, resulting, in theory, in more accurate topics .

Besides, on a more mathematical level, the assumptions made in the hierarchical structure of LDA are questionable. All topic-word distributions are assumed to be independent. It is really dependent on the data set whether this assumption is valid or not. If you look at the 'newsgroups' data set consisting of news articles, which is often used in the literature, it is natural to assume that topic-word distributions are independent, as they are very distinctive. However, for review data, this is less likely. There can be two topics that describe overlapping aspects or opinions, such that their topic-word distributions are not wholly distinctive nor independent making it harder to do inference.

9.1.2. Linguistics

Some aspects of language can not be taken into account in either basic LDA or the invented extension. A problematic word that quickly comes to mind is 'not'. This word says a lot about someone's opinion or sentiment on the product, while it is only considered a word, nothing more, nothing less. That is only if the word 'not' and a certain adjective or verb often occur together in a review, both words will be linked to a topic, and the sentiment of *'not + other word'* can be retrieved from the estimated topic-word distributions $\hat{\boldsymbol{\phi}}$. If this co-occurrence is not present, unfortunately, the sentiment or opinion linked to a topic will be inaccurate.

At the moment, in review analyses done by CQM, the phenomenon is counteracted by reading the top reviews belonging to each topic. Reading only a limited number of reviews saves time, while still, opinions corresponding to topics can be extracted. However, it would be better to improve LDA and its way to cope with words as 'not', such that the topic-word distributions themselves show the correct reviewers' opinions, and reading becomes superfluous.

Possibilities to incorporate the function of 'not' in an opinion in the data analysis are, for example, attaching the word 'not' to the previous or next word. This method is not very accurate, but easy to implement. Another option is to parse each sentence automatically (this can easily be done in Python using the package *nltk*), and retrieve the link of 'not' with the corresponding verb, adjective, adverb, or another part-of-speech in the sentence using the part-of-speech tagging. This is an implementable option, but it will take a lot of time to parse each sentence in each review. Besides, some reviews are not written in proper English, making accurately parsing more challenging. The last option is the incorporation of word order in the Latent Dirichlet Allocation model. This has already been done by some topic modellers (see for example [18], [17],[10]), and has been proved to work well. However, these models have increased complexity, and require thus, in general, more computation time and better programming skills.

9.1.3. Influence of the prior

In Bayesian statistics, we have seen that if the data set is large enough, the influence of the prior distribution will be negligible. Firstly, it is difficult to know when the data set is large enough to accurately estimate all latent random variables of interest. In general, whether the data set is large enough can be determined by changing the hyperparameters and checking the influence on the results. If this influence is null, the data set is large enough. If it is small and the main conclusions remain the same, the data set can be worked with, but if the results significantly change for different parameters, the data set is not large enough. Unfortunately, data sets cannot easily be increased, so the number of parameters should be decreased, leaving more relatively more data to estimate fewer parameters. Unfortunately, if the number of topics is decreased, it might be the case that two actual themes in the data set are assigned to the same topic. Therefore, the results must be interpreted more carefully in this case.

Furthermore, we have seen that the Adam optimization method is very sensitive to the values of the hyperparameters. This is an undesirable property, as the influence of the hyperparameters should vanish for large data sets. Therefore, Adam optimization as inference method is not preferred. Gibbs sampling shows a lot less sensitivity, so for the application of LDA to data, this method is recommended. Variational Bayesian EM is often used in LDA applications, but this inference method is above all advised against due to its independence assumptions that are considered too strong.

Given the fact that, in general, the data set is too small to make the influence of the prior disappear, we need to be considerate with choosing the prior distributions and the corresponding hyperparameters. Dirichlet distributions are good priors for Multinomial distributions, and Multinomials are appropriate for drawing categorical random variables such as topics or sentiments. Therefore, these prior distributions are a good fit for the topic modeling aim of Latent Dirichlet Allocation.

However, as mentioned before, the values of hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ can have a large influence on the results. Instead of choosing a symmetric $\boldsymbol{\alpha}$ of $\boldsymbol{\beta}$ hyperparameter vector, also an asymmetric prior can be chosen. In this way, one topic (in case of $\boldsymbol{\alpha}$) or a few words (in case of $\boldsymbol{\beta}$) are given preference in each document or each topic. Especially for the document-topic distributions, this can be wise, since it is not rare that every document is expected to go about one topic. This topic is then given preference beforehand in the chosen $\boldsymbol{\alpha}$ by assigning more mass to one dimension in the parameter vector of the Dirichlet distribution.

Because the asymmetric hyperparameter vector is still a powerful prior, an extra prior on $\boldsymbol{\alpha}$ can be imposed. A combination of extra priors on $\boldsymbol{\alpha}$ has been researched. One prior is imposed for the (a)symmetry, and another for the order of magnitude of $\boldsymbol{\alpha}$. The product then forms $\boldsymbol{\alpha}$, the hyperparameter for the document-topic distributions. As we have seen before, the order of magnitude of $\boldsymbol{\alpha}$ has a large influence on the way topics are distributed. A value larger than one for symmetric $\boldsymbol{\alpha}$ gives preference to a uniform distribution over the topics, while $\boldsymbol{\alpha}$ smaller than 1 gives preference to a distribution that assigns most mass to only one or a few dimensions. We want to learn this scaling aspect of $\boldsymbol{\alpha}$ from the data, and thus impose a prior on it. The supplementary (a)symmetry prior has been chosen to be a Dirichlet distribution with a parameter vector consisting of only ones, such that any possible document-topic distribution is equally probable. The distribution of the scaling prior is more difficult to choose, and several options have been thought of. Because we care about the order of magnitude, one might think of drawing $n \sim \text{Uniform}([-2,2])$, and then take 10^n for the order of magnitude. There are other possibilities for the distribution of n, as long as the resulting orders of magnitude are between 10^{-2} and 10^2. These are considered reasonable orders of magnitude for $\boldsymbol{\alpha}$. Because a double prior on $\boldsymbol{\alpha}$ causes the hierarchical model to become more complex concerning the form of the posterior density, this idea is not worked out more elaborately. It is however determined that with such a double prior, conjugacy in the Bayesian model is lost, and both Gibbs sampling and posterior mode determination via Adam optimization become computationally more challenging.

9.1.4. Model selection measures

We have seen in the results in chapter 8 that it is not easy to find a suitable measure to compare different models, whether the difference comes from different estimation methods, or different hyperparameters. Perplexity is a widely used measure based on information theory and is appropriate for topic models. However, it can only work with training and test sets, making it a lot more difficult to apply to LDA because the data set needs to be split. In this thesis, the split is performed within each document, such that approximately 80% of the words in a document is taken to train the model and estimate the required parameters, while 20 % is left for the test set. We have seen that the perplexity measure gives preference to models with flat distributions, that is each topic and each word have a relatively large probability on the document-topic and topic-word distributions respectively. However, if a topic does not occur in a document, it is preferred that the probability of that topic in the document-topic distribution is 0. Therefore, flat distributions are usually not good representations of the data. Numerically, document-topic probabilities being zero cause problems. As a consequence, 10^{-20} or an even smaller number is used instead. These numbers occur in the estimates of $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ by Adam optimization, but they react very poorly to the perplexity, resulting in a bad perplexity score.

One method to improve the perplexity score that can easily be implemented is the following. When computing the perplexity over a test set, words with low estimated probabilities in all topics can be ignored, such that their contribution to the perplexity is removed, whereas otherwise their small probabilities would have caused the perplexity to increase. Although this does not solve the entire problem, it is expected that the perplexity score will become more stable and less sensitive to low values in the topic-word distributions. Note that we do not wish to throw out topics with low probabilities in the same manner. If a topic has an overall low probability in all documents, we could better reduce the number of topics K. Other methods to improve the perplexity based on a different splitting the data sets into a training and test set are proposed in [48].

Besides the disappointing property of perplexity of dealing poorly with probabilities close to 0, it also does not correct for the number of parameters considered. This means that a lower and thus better perplexity score is reached for increasing K, the number of topics. However, for higher K, you might be overfitting, and accuracy of the topics decreases for an increasing number of parameters.

As an alternative, the value of the log posterior for the estimated random variables $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ is proposed. This measure does not suffer from the effects of training and test sets. The larger the log posterior value, the better the parameter estimations. This measure is based on the posterior mode being the best representation of the model. The aim of each inference method is then to get as close as possible to the posterior mode. This principle is comparable to maximum likelihood estimation in classical statistics. However, this measure can only be used to compare different methods of inference or different hyperparameters, since the number of parameters in total needs to be the same in each model. Whether the posterior mode indeed gives the best model parameters for LDA is not researched in this thesis. For annotated data sets, this phenomenon can be looked into by comparing the log posterior value for the actual, true model parameters with the posterior mode.

Apart from the perplexity and the log posterior value, measures for better model comparison need to be found. Fortunately, the data in LDA consists of words, such that through reading, when can get qualitatively validate the correctness of the model parameters. This becomes however more difficult for large data sets with many parameters. It is better to find a quantitative measure with a valid mathematical argumentation to conclude on which inference method is the best for LDA, and to find the best settings for K and hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. More research can thus be done on good model comparison and model validation measures for topic modeling.

9.2. LDA with syntax and sentiment

The extension to LDA using sentences and sentiments has a good theoretical foundation. The assumption that in each phrase, only one topic is described with one sentiment is reasonable, although not always true. Also, the manner of splitting a review into sentences can be discussed, as rules do not apply to each review. Think of splitting on the word ‘and’ for example. This word can either be part of a summation, in which it is not always informative to split on *‘and’*, or can be a conjunction between two main phrases. In the latter case, it is wise to split on ‘and’ because a complete phrase exists on both sides of the word. The same can be said about splitting on commas. Therefore, it is recommended to read a small subset of the (review) data set to get a grasp of the writing style that is used, such that, with this knowledge, you can decide which splitting rules are the most appropriate.

Furthermore, for the assignment of positive and negative sentiment to a review, word lists with the most common positive and negative words in the English language are used. These word lists are not perfect, and it will be the case that some words are classified to a sentiment incorrectly. In further applications, these word lists need to be perfected and adapted according to the used data set. Words that are positive in one context, might be negative in another, think of, e.g., ‘close’.

Lastly, Adam optimization is not robust to hyperparameter settings and initialization. It can find the parameters of a simulated data set, but a remark needs to be made there. The simulated data set consisted of very distinct topic-sentiment-word distributions, making it easier to find the correct parameters. We have seen for the stroller data, that is a lot more difficult to find the right settings and interpretable estimates.
Furthermore, the algorithm only works on small data sets in the current implementation. More research can be done on the both the robustness and the upscaling of Adam optimization for determining the posterior mode estimates of interest in LDA with syntax and sentiment.
Besides, in the comparison of inference methods for plain LDA, we have seen that Gibbs sampling performs better than Adam optimization in terms of parameter estimation. Therefore, it is expected that Gibbs sampling for LDA with syntax and sentiment will give us better results, with which hopefully manually reading becomes superfluous.

10

Conclusion and recommendations

The main goal of this work is to find good methods to summarize customers' opinions in review data, without having to read all reviews. To this end, Latent Dirichlet Allocation has been described, and various inference methods to estimate its model parameters have been researched. Moreover, a new method of inference, Adam optimization, has been applied to LDA for two different data sets.
Because this thesis is written in collaboration with CQM, an extension to LDA that is more suitable for review analyses and opinion mining is invented, called LDA with syntax and sentiment. For this new topic model, only one inference method is tested on a simulated data set, and a real data set. The main conclusions will be given in the first section of this chapter. Subsequently, recommendations are proposed for further research.

10.1. Main findings

Latent Dirichlet Allocation is a hierarchical Bayesian topic model that allocates topics to words, and finds in this way topic distributions per document and word distributions per topic. From the document-topic distributions, we can conclude which topics occur most frequently in the reviews. The topic-word distributions give insights in what the topics are about, such that we know more about the customers' opinions on the specific product they describe in the set of reviews. Marketing strategies and product innovations can then be steered into a specific direction using the conclusions from the topics in the review data set.

Three different inference methods are used to estimate the document-topic and topic-word distributions in LDA: Gibbs sampling, Variational Bayesian Expectation-Maximization, and Adam optimization. From these methods, Gibbs sampling using the application in the program KNIME gives the most robust and best results. The inference method which is new concerning its application for LDA, Adam optimization, disappoints, since it is sensitive to hyperparameter settings and initialization. Only for hyperparameters larger than 1, a setting that is only preferred in a limited number of cases, the algorithm performs well.

After having estimated the parameters in the topic-word distributions, a check is needed to determine whether the chosen number of topics was correct. That is if the number of topics is too large, the model overfits, which results in some topics consisting of noise. This noise is not informative and should not be interpreted. Therefore, two similarity measures are computed for all topic-word distribution combinations, the normalized symmetric Kullback-Leibler divergence and the normalized symmetric Jensen-Shannon divergence. Both perform well and can thus be used to check for overfitting in terms of the number of topics.

10.2. Further research

Although in this thesis different methods of inference are looked into, a new one is proposed, and even an new topic model is introduced, research is never finished. Therefore, we will shortly discuss recommendations for further research in this section.

Firstly, better model validation and model comparison measures need to be defined. Currently, in the field of

topic modeling, the perplexity is used. For this measure, a division of the data set in a training and test set is required. However, model parameters are estimated on document-level, with as result that this split is not straightforward. More research can be done on methods to form a training and test set for the computation of the perplexity. Moreover, the perplexity has two disadvantages that are both inherent to its definition: it has a preference for flat distributions and for overfitted models. Lastly, it says something about the predictive power of LDA, while the aim of LDA in this research is to summarize data sets.
Another possible method to compare model outcomes is the log posterior value. This is the value of the posterior density (up to a proportionality constant) for the estimated model parameter. It is essential that the parameters of the models that are compared have the same dimensions. This is a downfall, as it would be ideal if the model comparison score could give insights in the best number of topics to use. Furthermore, it is not certain that the highest log posterior value, that is, the posterior mode, also gives the best estimates of the model parameters for LDA. More research can be done whether or not this is true for Latent Dirichlet Allocation.
Because both model validation scores do not perform satisfactorily, a better score needs to be thought of. The outcomes of topic models are often interpreted qualitatively, thus, a score in which human interpretation of text can be combined with automatic reading is preferred.

Secondly, Adam optimization has been applied to the posterior density of the Latent Dirichlet Allocation model to find the posterior mode. This method can only find the posterior mode and estimate model parameter correctly for specific hyperparameter settings. That is for hyperparameters smaller than 1, the algorithm walks towards the boundaries of the domain, with as result that optimization is only performed in a limited number of dimensions. A method to prevent the algorithm from showing this behavior can be researched. Either the algorithm can be improved, or another way of making the model prefer documents having only a few topics by using, for example, an extra prior can be thought of.

Lastly, the extended version of LDA, specifically designed for review analyses, can be perfected. The theoretic framework underlying LDA with syntax and sentiment is realistic, but Adam optimization as inference method leaves much to be desired. Adam optimization is used to search for the posterior mode estimates for the document-topic, document-sentiment, and topic-sentiment-word distributions, but it is challenging to find the good settings in the algorithm, and the method is very sensitive to initialization. Only for a simulated data set with distinct topic-sentiment-word distributions, good estimates have been found.
Already for basic LDA, it had been shown that Adam optimization does not work as desired. Therefore, it is recommended to apply Gibbs sampling to LDA with syntax and sentiment to find estimates for the model parameter. Still, much is expected from the new extension to LDA, only the best method of inference needs to be discovered.

After all, research is nothing more than searching for answers to your problems, and consequently adapting the questions themselves. Following the wise words of Sherlock Holmes in the works of Sir Arthur Conan Doyle (1859-1930):

> *"Once you eliminate the impossible, whatever remains, no matter how improbable, must be the truth."*

A

Mathematical background and derivations

A.1. Functional derivative and Euler-Lagrange equation

In the derivation of the variational Bayesian EM algorithm with the mean field approximation, a functional derivative is used to determine the form of the auxiliary distribution for which the functional, in this case a lower bound for the log likelihood, is maximal. In this section, the functional derivative will be derived and the definition of the differential (derivative of functional) is given. But first some general definitions and result from functional analysis are stated.

Let X be a vector space, Y a normed space and T a transformation defined on a domain $D \subset X$ and having range $R \subset Y$. [29]

Definition A.1 (Gateaux differential)
Let $x \in D \subset X$ and let h be arbitrary in X. If the limit

$$\delta T(x;h) = \lim_{\alpha \to 0} \frac{1}{\alpha} [T(x+\alpha h) - T(x)] \tag{A.1}$$

exists, it is called the Gateaux differential of T at x with increment h. If the limit in A.1 exists for each $h \in X$, the transformation T is said to be Gateaux differentiable at x.

A more frequently used definition of the Gateaux differential is the following [29]: if f is a functional on X, the Gateaux differential of f, if it exists, is

$$\delta f(x;h) = \frac{d}{d\alpha} f(x+\alpha h)\Big|_{\alpha=0} \tag{A.2}$$

and for each fixed $x \in X$, $\delta f(x;h)$ is a functional with respect to the variable $h \in X$.

A stronger differential is the Fréchet differential, which is defined on a normed space X. This differential enhances continuity. [29]

Definition A.2 (Fréchet differential)
Let T be a transformation defined on an open domain D in a normed space X and having range in a normed space Y. If for fixed $x \in D$ and each $h \in X$, there exists $\delta T(x;h) \in Y$ which is linear and continuous with respect to h such that

$$\lim_{\|h\| \to 0} \frac{\| T(x+h) - T(x) - \delta T(x;h) \|}{\|h\|} = 0 \tag{A.3}$$

then T is said to be Fréchet differentiable at x and $\delta T(x;h)$ is said to be the Fréchet differential of T at x with increment h.

Three general propositions from [29] are summarized in proposition A.1. For the proofs, we refer to [29].

Proposition A.1 (Properties Fréchet differential)
The following are true for a transformation T defined on an open domain D in a normed space X and having range in a normed space Y:

- *If the transformation T has a Fréchet differential, it is unique.*
- *If the Fréchet differential of T exists at x, then the Gateaux differential exists at x and they are equal.*
- *If the transformation T defined on an open set D in X has a Fréchet differential at x, then T is continuous at x.*

Now consider a functional $\mathscr{F}$ of the form:

$$\mathscr{F} = \int_{x_1}^{x_2} L(q(x), \dot{q}(x), x)\, dx \tag{A.4}$$

Where $\dot{q}(x) = \frac{dq}{dx}$.

A classical problem in the field of variational calculus is finding a function q on $[x_1, x_2]$ that minimizes the functional $\mathscr{F}$ [29]. The admissible set of functions for this problem consists of all functions that are continuous and whose derivatives are continuous in the range $[x_1, x_2]$. Besides let q be an admissible function and suppose there exists h such $q+h$ is also admissible. All such possible functions h are collected in the class of so-called admissible variations. Also restrict the set of admissible functions to those whose end points i.e. $q(x_1)$ and $q(x_2)$ are fixed.

The Gateaux differential of functional $\mathscr{F}$, assume that it exists, is given by:

$$\begin{aligned} \delta\mathscr{F}(q;h) &= \frac{d}{d\alpha}\int_{x_1}^{x_2} L(q+\alpha h, \dot{q}+\alpha\dot{h}, x)\, dx\,\Big|_{\alpha=0} \\ &= \int_{x_1}^{x_2} L_q(q,\dot{q},x)h(x)\, dx + \int_{x_1}^{x_2} L_{\dot{q}}(q,\dot{q},x)\dot{h}(x)\, dx \end{aligned} \tag{A.5}$$

It can be verified that this differential is also Fréchet [29]. Now, theorem A.1 gives a necessary condition for the extrema of functional $\mathscr{F}$, as desired.[29]

Theorem A.1 (Extrema of a functional)
Let the real-valued function f have a Gateaux differential on a vector space X. A necessary condition for f to have an extremum at $x_0 \in X$ is that $\delta f(x_0; h) = 0$ for all $h \in X$.

As proposition A.1 indicates that every Fréchet differential is also a Gateaux differential, we can apply theorem A.1 to equation A.5 to find the extremum.

$$\begin{aligned} \delta\mathscr{F}(q;h) &= \int_{x_1}^{x_2} L_q(q,\dot{q},x)h(x)\, dx + \int_{x_1}^{x_2} L_{\dot{q}}(q,\dot{q},x)\dot{h}(x)\, dx = 0 \\ &= \int_{x_1}^{x_2} \left[L_q(q,\dot{q},x)h(x) + L_{\dot{q}}(q,\dot{q},x)\dot{h}(x)\right] dx = 0 \end{aligned} \tag{A.6}$$

To arrive from the equation above to the Euler-Lagrange equation, we use one of the fundamental lemmas from variational calculus [29]:

Lemma A.1
If $\alpha(t)$ and $\beta(t)$ are continuous in $[t_1, t_2]$ and

$$\int_{t_1}^{t_2} \left[\alpha(t)h(t) + \beta(t)\dot{h}(t)\right] dt = 0 \tag{A.7}$$

for every $h \in D[t_1, t_2]$ with $h(t_1) = h(t_2) = 0$, then β is differentiable and $\dot{\beta}(t) \equiv \alpha(t)$ in $[t_1, t_2]$.

Note that in equation A.6, we have the same form as in lemma A.1. Therefore:

$$\begin{aligned} L_q(q,\dot{q},x) &= \frac{d}{dt} L_{\dot{q}}(q,\dot{q},x) \\ \Rightarrow L_q(q,\dot{q},x) - \frac{d}{dt} L_{\dot{q}}(q,\dot{q},x) &= 0 \end{aligned} \tag{A.8}$$

This last result in known as the Euler-Lagrange equation, which is used in this thesis for the derivation of the variational Bayesian EM update equations.

A.2. Expectation of logarithm of Beta distributed random variable

In the derivation of the update equations in variational Bayesian EM for general LDA, the expectation of the logarithm of a Beta distributed random variable occurs several times. In this section its computation will be elaborated on. Consider the latent random vector $\boldsymbol{\Theta}$ which is Dirichlet distributed with parameter vector $\boldsymbol{\alpha}$. From previous results, we know that $(\boldsymbol{\Theta})_i$ is Beta distributed with parameters $(\boldsymbol{\alpha})_i$ and $\sum_{j\neq i}(\boldsymbol{\alpha})_j$. The probability density function of $(\boldsymbol{\Theta})_i$ is therefore:

$$\begin{aligned}
p((\boldsymbol{\theta})_i|(\boldsymbol{\alpha})_i,\sum_{j\neq i}(\boldsymbol{\alpha})_j) &= \frac{\Gamma\left(\sum_{k=1}^K(\boldsymbol{\alpha})_k\right)}{\Gamma((\boldsymbol{\alpha})_i)\cdot\Gamma\left(\sum_{j\neq i}(\boldsymbol{\alpha})_j\right)}\cdot(\boldsymbol{\theta})_i^{(\boldsymbol{\alpha})_i-1}\cdot(1-(\boldsymbol{\theta})_i)^{\sum_{j\neq i}(\boldsymbol{\alpha})_j-1}\\
&= \exp\left\{((\boldsymbol{\alpha})_i-1)\log((\boldsymbol{\theta})_i)+(\sum_{j\neq i}(\boldsymbol{\alpha})_j-1)\log(1-(\boldsymbol{\theta})_i)+\log\left(\Gamma\left(\sum_{k=1}^K(\boldsymbol{\alpha})_k\right)\right)-\log\left(\Gamma\left(\sum_{j\neq i}(\boldsymbol{\alpha})_j\right)\right)-\log(\Gamma((\boldsymbol{\alpha})_i))\right\}\\
&= h((\boldsymbol{\theta})_i)\cdot\exp\left\{\eta_1 t_1((\boldsymbol{\theta})_i)+\eta_2 t_2((\boldsymbol{\theta})_i)-A(\eta)\right\}
\end{aligned} \tag{A.9}$$

From equation A.9 we can conclude that the distribution of $(\boldsymbol{\Theta})_i$ belongs to an exponential family with natural statistics $\log((\boldsymbol{\theta})_i)$ and $\log(1-(\boldsymbol{\theta})_i)$ and natural parameters $(\boldsymbol{\alpha})_i-1$ and $\sum_{j\neq i}(\boldsymbol{\alpha})_j$. The normalization constant is given by $A(\eta)$. Now we can use some useful results from exponential family distribution, namely its moment generating function, which we will derive for the first moment as follows.
First note that:

$$e^{A(\eta_1,\eta_2)} = \int h((\boldsymbol{\theta})_i)\cdot\exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{t}((\boldsymbol{\theta})_i)\right\}d(\boldsymbol{\theta})_i \tag{A.10}$$

Differentiating both sides with respect to η gives:

$$\begin{aligned}
\nabla_\eta e^{A(\eta_1,\eta_2)} &= \nabla_\eta\left(\int h((\boldsymbol{\theta})_i)\cdot\exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{t}((\boldsymbol{\theta})_i)\right\}d(\boldsymbol{\theta})_i\right)\\
e^{A(\eta_1,\eta_2)}\nabla_\eta A(\eta_1,\eta_2) &= \left(\int h((\boldsymbol{\theta})_i)t_1((\boldsymbol{\theta})_i)\exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{t}((\boldsymbol{\theta})_i)\right\}d(\boldsymbol{\theta})_i\ ,\ \int h((\boldsymbol{\theta})_i)t_2((\boldsymbol{\theta})_i)\exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{t}((\boldsymbol{\theta})_i)\right\}d(\boldsymbol{\theta})_i\right)\\
\Rightarrow \nabla_\eta A(\eta_1,\eta_2) &= (\mathbb{E}\left[t_1((\boldsymbol{\theta})_i)\right],\mathbb{E}\left[t_2((\boldsymbol{\theta})_i)\right])
\end{aligned} \tag{A.11}$$

Where integration and differentiation can be interchanged via dominated convergence. The result is that we have obtained expressions for the expectations of each natural statistic. For the derivation of the variational Bayesian EM algorithm for LDA, the expectation of $\log(\Theta)$ was needed, which is also one of the two natural statistics. Therefore:

$$\begin{aligned}
\mathbb{E}\left[\log((\boldsymbol{\theta})_i)\right] = \frac{\partial A(\eta_1,\eta_2)}{\partial\eta_1} &= \frac{\partial}{\partial((\boldsymbol{\alpha})_i-1)}\left(-\log\left(\Gamma\left(\sum_{k=1}^K(\boldsymbol{\alpha})_k\right)\right)+\log\left(\Gamma\left(\sum_{j\neq i}(\boldsymbol{\alpha})_j\right)\right)+\log(\Gamma((\boldsymbol{\alpha})_i))\right)\\
&= -\Psi\left(\Gamma(\sum_{k=1}^K(\boldsymbol{\alpha})_k)\right)+\Psi\left((\boldsymbol{\alpha})_i\right)
\end{aligned} \tag{A.12}$$

With $\Psi(\cdot)$ being the digamma function.

A.3. LDA posterior mean determination

One of the possibilities to estimate parameters using Bayesian statistics is to compute the posterior mean. In this section, the determination of the posterior mean for the high-dimensional hierarchical Bayesian model that LDA is, is derived.

Suppose we start with the computation of the posterior mean for each $\mathbf{\Theta_d}$. To retrieve an expression for this estimator from the joint posterior distribution $p(\boldsymbol{\theta},\boldsymbol{\phi}|\mathbf{w})$, we can first condition on $\boldsymbol{\phi}$ and then compute the mean over $\mathbf{\Phi}$. In formulas:

$$\mathbb{E}[\mathbf{\Theta}|\mathbf{w}] = \mathbb{E}_{\mathbf{\Phi}}\left[\mathbb{E}_{\mathbf{\Theta}|\mathbf{\Phi}}\left[\mathbf{\Theta}|\mathbf{\Phi},\mathbf{w}\right]\right] \tag{A.13}$$

The subscript in $\mathbb{E}_{\mathbf{\Phi}}$ denotes the fact that the expectation is taken with respect to random variable $\mathbf{\Phi}$. Because each $\mathbf{\Theta_d}$ is independent of the topic distributions of other documents, $\mathbf{\Theta_j}$ for $j \neq d$, the posterior mean for $\mathbf{\Theta_d}$ can be computed separately for each document. Note that we still need to condition on all topic-word distributions $\mathbf{\Phi_k}$ for $k = 1,\ldots,K$.

$$\mathbb{E}\left[\mathbf{\Theta_d}|\mathbf{w}\right] = \mathbb{E}_{\mathbf{\Phi}}\left[\mathbb{E}_{\mathbf{\Theta_d}|\mathbf{\Phi}}\left[\mathbf{\Theta_d}|\mathbf{\Phi},\mathbf{w}\right]\right] \tag{A.14}$$

To this end, we need an expression for the conditional posterior of $\mathbf{\Theta_d}$ given $\mathbf{\Phi}$ and $\mathbf{w}$. Using equation 4.2, we arrive at:

$$\begin{aligned} p(\boldsymbol{\theta}|\boldsymbol{\phi},\mathbf{w}) &\propto \left[\prod_{d=1}^{M}\prod_{j=1}^{V}\left(\sum_{k=1}^{K}(\boldsymbol{\phi}_\mathbf{k})_j(\boldsymbol{\theta}_\mathbf{d})_k\right)^{n_{d,j}}\right]\cdot\left[\prod_{d=1}^{M}\prod_{k=1}^{K}(\boldsymbol{\theta}_\mathbf{d})_k^{(\boldsymbol{\alpha})_k-1}\right] \\ &= \prod_{d=1}^{M}\left\{\left[\prod_{j=1}^{V}\left(\sum_{k=1}^{K}(\boldsymbol{\phi}_\mathbf{k})_j(\boldsymbol{\theta}_\mathbf{d})_k\right)^{n_{d,j}}\right]\cdot\left[\prod_{k=1}^{K}(\boldsymbol{\theta}_\mathbf{d})_k^{(\boldsymbol{\alpha})_k-1}\right]\right\} \\ p(\boldsymbol{\theta}_\mathbf{d}|\boldsymbol{\phi},\mathbf{w}) &\propto \left[\prod_{j=1}^{V}\left(\sum_{k=1}^{K}(\boldsymbol{\phi}_\mathbf{k})_j(\boldsymbol{\theta}_\mathbf{d})_k\right)^{n_{d,j}}\right]\cdot\left[\prod_{k=1}^{K}(\boldsymbol{\theta}_\mathbf{d})_k^{(\boldsymbol{\alpha})_k-1}\right] \end{aligned} \tag{A.15}$$

The distribution of $(\mathbf{\Theta_d}|\mathbf{\Phi},\mathbf{w})$ in equation A.15 is called the generalized Dirichlet distribution and is defined in [11] as follows.

Definition A.3 (Generalized Dirichlet distribution)
Let $\mathbf{u}$ *and* $\mathbf{b}$ *be* K*-dimensional vectors, let* Z *be a* $[K\times\kappa]$ *matrix and let* $\boldsymbol{\beta}$ *be a* κ*-dimensional vector. If* $\mathbf{u}$ *is distributed as* $\mathrm{Dir}^{\kappa}(\mathbf{b},\mathrm{Z},\boldsymbol{\beta})$*, then its probability density function defined on the* $(K-1)$*-simplex is given by:*

$$f(\mathbf{u};\mathbf{b},\mathrm{Z},\boldsymbol{\beta}) = \frac{B(\mathbf{b})^{-1}\left(\prod_{i=1}^{K}u_i^{b_i-1}\right)\left[\prod_{j=1}^{\kappa}\left(\sum_{i=1}^{K}u_i z_{i,j}\right)^{-\beta_j}\right]}{\mathscr{R}_{-\beta}(\mathbf{b},\mathrm{Z},\boldsymbol{\beta})} \tag{A.16}$$

Where $\mathscr{R}$ *is a double Dirichlet average [11], which is explained in definition A.4 below.*

Therefore we know that $\left(\mathbf{\Theta_d}|\boldsymbol{\phi},\mathbf{w}\right) \sim \mathrm{Dir}^{V}(\boldsymbol{\alpha},\Phi,-\mathbf{n_d})$, where matrix Φ consists of all vectors $\boldsymbol{\phi}$ concatenated row-wise. That is, matrix element $\Phi_{i,j}$ is the j-th element of topic-word distribution vector $\boldsymbol{\phi}_\mathbf{i}$.

Definition A.4 (Double Dirichlet average)
The double Dirichlet average is the generalization of the function R*, Carlson's multiple hypergeometric function. Consider a matrix* Z $(K\times\kappa)$*, vectors* $\mathbf{u}$ *and* $\mathbf{b}$ *of size* $K\times 1$ *and vectors* $\mathbf{v}$ *and* $\boldsymbol{\beta}$ *of size* $\kappa\times 1$*. The double Dirichlet average for some* a *is then defined as follows:*

$$\begin{aligned} \mathscr{R}_a(\mathbf{b},Z,\boldsymbol{\beta}) &= \mathbb{E}_{\mathbf{u}|\mathbf{b}}\left[\mathbb{E}_{\mathbf{v}|\boldsymbol{\beta}}\left[(\mathbf{u}^T Z\mathbf{v})^a\right]\right] \\ &= \mathbb{E}_{\mathbf{u}|\mathbf{b}}\left[R_a(\boldsymbol{\beta};Z^T\mathbf{u})\right] \end{aligned} \tag{A.17}$$

For $a = -\beta.$*, where* $\beta. = \sum_i \beta_i$*, the double average becomes [11]:*

$$\mathscr{R}_{-\beta}(\mathbf{b},Z,\boldsymbol{\beta}) = \mathbb{E}_{\mathbf{u}|\mathbf{b}}\left[\prod_{j=1}^{K}\left(\sum_{i=1}^{K}u_i\cdot z_{i,j}\right)^{-\beta_j}\right] \tag{A.18}$$

In [20], Jiang et al. present several methods to approximate Carlson's $\mathscr{R}$ and they derive its exact computation for certain special cases.

In order to obtain the posterior mean of $\boldsymbol{\Theta}$, we need to know the mean of the distribution of $\left(\boldsymbol{\Theta}|\boldsymbol{\phi},\mathbf{w}\right)$ first. Therefore an expression for the product moment function of a generalized Dirichlet distribution could be used.

Proposition A.2 (Product moment generalized Dirichlet distributed random variable)
Consider the case in definition A.3, that is $\mathbf{u} \sim \mathrm{Dir}^{\kappa}(\mathbf{b}, Z, \boldsymbol{\beta})$. *Then its product moment for vectors* $\mathbf{m}$ *and* $\boldsymbol{\mu}$ *is:*

$$\mathbb{E}_{\mathbf{u}|\mathbf{b},Z,\boldsymbol{\beta}}\left[\left(\prod_{i=1}^{K} u_i^{m_i}\right)\prod_{j=1}^{\kappa}\left(\sum_{k=1}^{K} u_i \cdot z_{i,j}\right)^{-\mu_j}\right] = \frac{B(\mathbf{b}+\mathbf{m})}{B(\mathbf{b})}\frac{\mathscr{R}_{-(\beta+\mu)}(\mathbf{b}+\mathbf{m}, Z, \boldsymbol{\beta}+\boldsymbol{\mu})}{\mathscr{R}_{-\beta}(\mathbf{b}, Z, \boldsymbol{\beta})} \tag{A.19}$$

In the case of the document-topic distribution in LDA, we know that $\left(\boldsymbol{\Theta}_{\mathbf{d}}|\boldsymbol{\phi},\mathbf{w}\right) \sim \mathrm{Dir}^{V}(\boldsymbol{\alpha}, \Phi, \mathbf{n_d})$, with $\boldsymbol{\theta}_{\mathbf{d}}$ the topic distribution of document d, $\boldsymbol{\alpha}$ the prior parameter vector, Φ the matrix with topic-word probabilities having size $(K \times V)$ and $\mathbf{n_d}$ the vector with all word occurrences (in counts) in document d. With this analytical expression for the conditional distribution, we can compute the conditional posterior mean in closed form, using equation A.19. Taking $\mathbf{m} = \mathbf{i}$ (with $\mathbf{i}$ the unit vector i.e. $(0,0,\ldots,1,0,\ldots,0)$ with 1 in the i-th place) and $\boldsymbol{\mu} = \mathbf{0}$ in equation A.19:

$$\mathbb{E}\left[(\boldsymbol{\Theta}_{\mathbf{d}})_i|\boldsymbol{\Phi},\mathbf{w}\right] = \frac{B(\boldsymbol{\alpha}+\mathbf{i})}{B(\boldsymbol{\alpha})}\frac{\mathscr{R}_{-N_d}(\boldsymbol{\alpha}+\mathbf{i}, \Phi, \mathbf{n_d})}{\mathscr{R}_{-N_d}(\boldsymbol{\alpha}, \Phi, \mathbf{n_d})} \tag{A.20}$$

As a result, there is an analytical expression for the conditional posterior mean of $(\boldsymbol{\Theta}_{\mathbf{d}})_i$ for each document d and each dimension $i = 1,\ldots,K$. However, in order to obtain a general, unconditional posterior mean for $(\boldsymbol{\Theta}_{\mathbf{d}})_i$, the expected value of expression A.20 with respect to Φ needs to be determined. Therefore, we need to integrate over the ratio of double Dirichlet averages $\mathscr{R}$. To this end, it is better to have a simplified expression for $\mathscr{R}(\boldsymbol{\alpha}+\mathbf{i}, \Phi, \mathbf{n_d})$, such that we might recognize a probability distribution of which an exact form of the mean is known or equation A.20 can be written such that we can compute the integral.

Jiang et al. propose in [20] two methods to compute the double Dirichlet average analytically and two methods that approximate its value. For the first two methods, matrix Φ needs to satisfy several assumptions, which are true for LDA only in rare cases. The first method, in which $\mathscr{R}$ can be calculated relatively easily, requires that matrix Φ must be a n-level nested partition indicator matrix. A n-level nested partition indicator matrix is a matrix whose columns are indicator vectors of the n-level nested partition subsets. The indicator vectors are vectors that take on value 1 for index i if category i is in the subset and value 0 otherwise. The n-level nested partition subsets are explained more thoroughly in [20], but one can think of these sets as forming a partition of the set $\{1,\ldots,V\}$ while either being subsets of each other or being disjoint. That is, the subsets in the n-level nested partition cannot partially overlap. Note that in the case of LDA, matrix Φ is the considered matrix in $\mathscr{R}$, which is a probability matrix with values in the interval $[0,1]$ summing row-wise to 1. Therefore it can only satisfy the requirement for exact computation if each topic-word distribution gives all probability to one word and 0 probability to all other words in the vocabulary. Only then the matrix will consist of 0's and 1's, as required in this method. Because this is not a realistic case for the considered model in this thesis, the first method of exact computation of the double Dirichlet average is discarded.

Secondly, the so-called 'expansion method' is proposed in [20]. This method is valid for any matrix Φ and uses the fact that vector $\mathbf{n_d}$ consists of non-negative integers. This results in a simplification of $\mathscr{R}(\boldsymbol{\alpha}+\mathbf{i}, \Phi, \mathbf{n_d})$ via the introduction of matrix W. For the exact procedure, we will refer to the explanation in [20]. This method does result in an analytical expression for $\mathscr{R}$, however, we need to sum over all possible matrices W, which results in a sum over $\prod_{u=1}^{U}((\mathbf{n_d})_u + 1)$ terms, where U is the number of unique words in document d and $(\mathbf{n_d})_u$ the frequency of word u in document d. Furthermore, note that the entries of matrix Φ are unknown, as they are still random variables of which the expectation needs to be computed. Although analytically it is possible to write out all possibilities for W and get an expression of the double Dirichlet average $\mathscr{R}$ for each possible W in terms of beta functions and Φ, the method is considered computationally intractable. This conclusion is also drawn in [20] for high dimensional data sets, which is the case in this project.

The first approximation method of Jiang et al. is the application of Laplace's approximation. In order to be allowed to apply this formula, we need the function:

$$g(\boldsymbol{\theta}_\mathbf{d}) = \prod_{j=1}^{J} \left(\sum_{k=1}^{K} (\boldsymbol{\Phi}_\mathbf{k})_j \cdot (\boldsymbol{\Theta}_\mathbf{d})_K \right)^{m_j} \tag{A.21}$$

to have one single mode. Requirements for this condition to be true are that all terms m_j are strictly positive and the columns of Φ must span the K-dimensional vector space. The first requirement is not always true, as in vector $\mathbf{m}$ the word counts for document d, $\mathbf{n_d}$, are included and it is not seldom that a word in the vocabulary does not occur in the specific document d such that $n_{d,j} = 0$ for word j. The second requirement that the columns of Φ span the K-dimensional vector space is not verifiable beforehand, as Φ is a random matrix whose values are unknown. Therefore it cannot be guaranteed that function $g(\boldsymbol{\theta}_\mathbf{d})$ has a single mode, which we already expected due to the possibility of topic permutations. With this being a strict requirement for the Laplace approximation, this method can unfortunately not be applied to the double Dirichlet averages in equation A.20.

At last, a Monte Carlo method is proposed to determine $\mathscr{R}$. Here the fact that $\mathscr{R}$ is actually the mean of a function of the form in equation A.21, as stated in equation A.18 in the definition of the double Dirichlet average, is used. Therefore, given matrix Φ, we could simulate the Dirichlet process by drawing x_i from a gamma($(\boldsymbol{\alpha})_i, 1$) distribution for $i = 1, \ldots, K$ (see section 2.2.1), then computing $u_i = \frac{x_i}{\sum_j x_j}$ and substituting this in equation A.18. Note that vector $\boldsymbol{\beta}$ in equation A.18 is known and we have assumed that matrix Φ is also known. However, as aforementioned, Φ is a random matrix whose values are unknown, so the Monte Carlo procedure to determine $\mathscr{R}$ cannot be executed. Naturally, we could take different numerical examples for Φ and use these in the approximation. Only there are infinitely many options for matrix Φ, as each element can take on any value in $[0,1]$ as long as they sum row-wise to 1. It is therefore considered unfeasible to use this method for the approximation of $\mathscr{R}$.

Research on approximations of Carlson's multiple hypergeometric function $\mathscr{R}$ is mostly focused on problems in a Bayesian setting, categorical data and missing values, see for example [12]. In those cases, the matrix in $\mathscr{R}$ is a n-level nested partition indicator matrix or can be transformed to one. Consequently, the posterior mean can be determined analytically. The writer of this thesis has not found other methods than those proposed in [20] to determine $\mathscr{R}$ and therefore the posterior mean of $\boldsymbol{\theta}_\mathbf{d}$ is considered both computationally and in most cases also analytically intractable. That is, computations are too exhaustive and in most cases no analytical expression for the posterior mean is known.

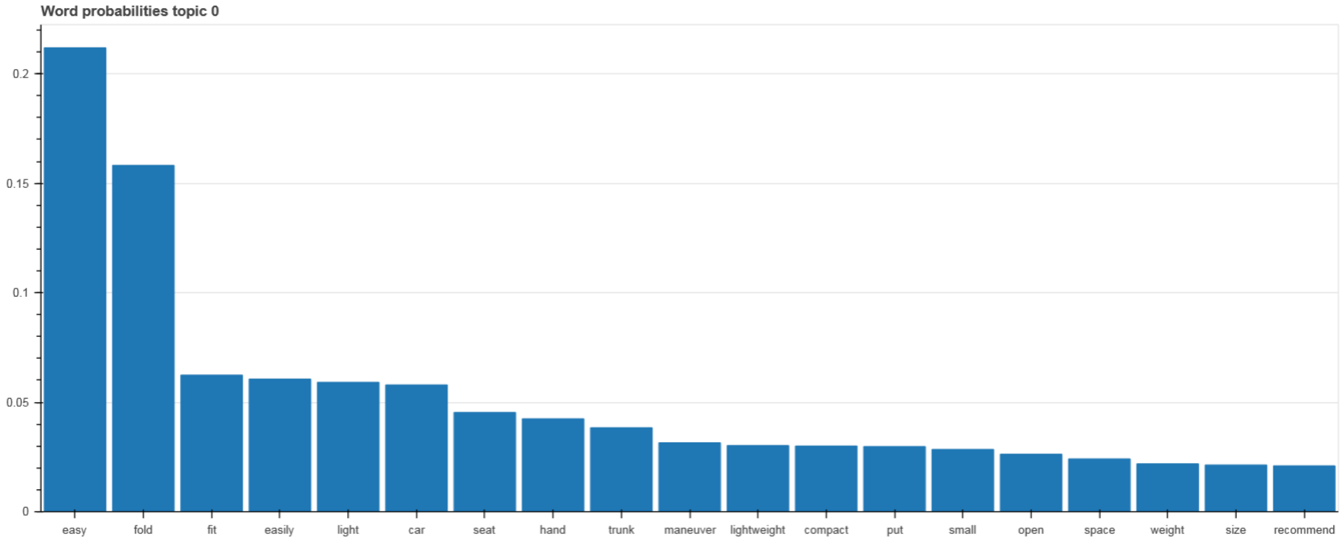
Besides, when we look at the posterior mean for Φ, which naturally also needs to be determined, we see that the conditional posterior distribution of $(\boldsymbol{\Phi}|\boldsymbol{\theta}, \mathbf{w})$ is not even a generalized Dirichlet distribution. Computations are expected to be even more difficult in this case, so the same conclusion can be drawn.

In conclusion, the posterior mean of the desired parameters cannot be determined analytically, therefore we will need to resort to approximation methods for the posterior distribution in order to compute the posterior mean.

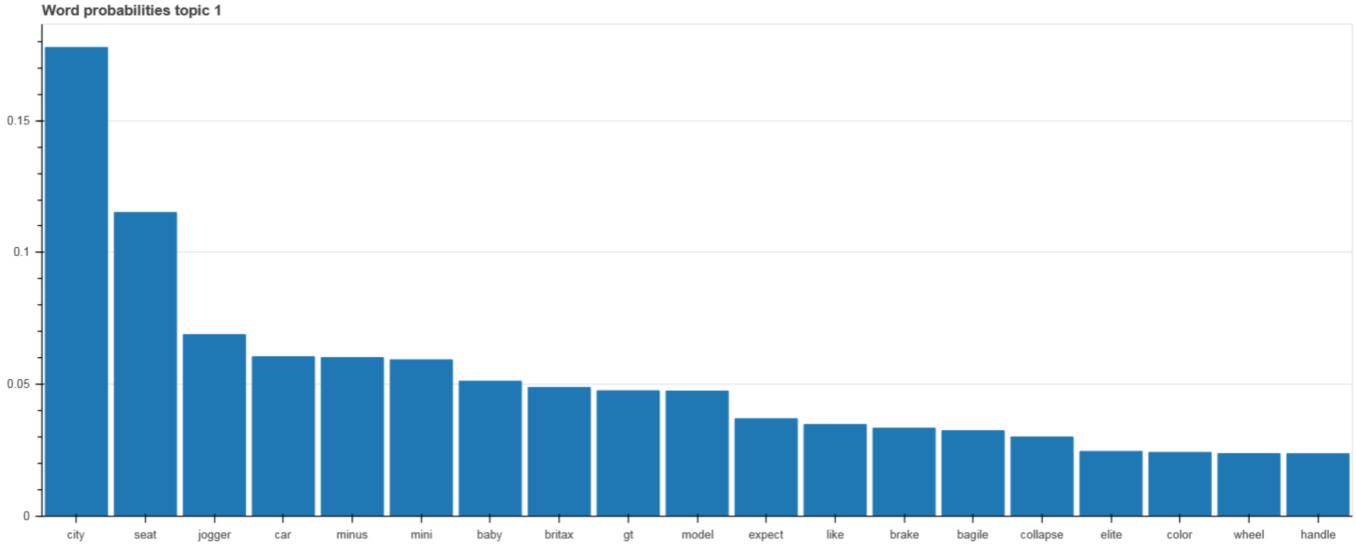
B

Results and data sets

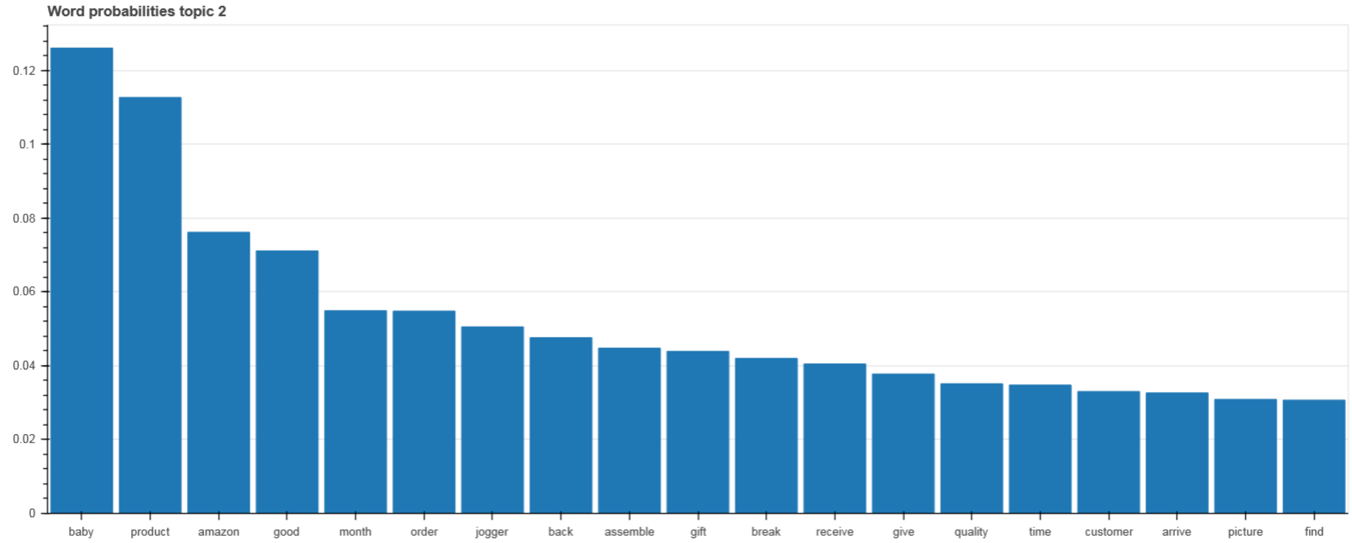
B.1. Stroller topic-word distributions

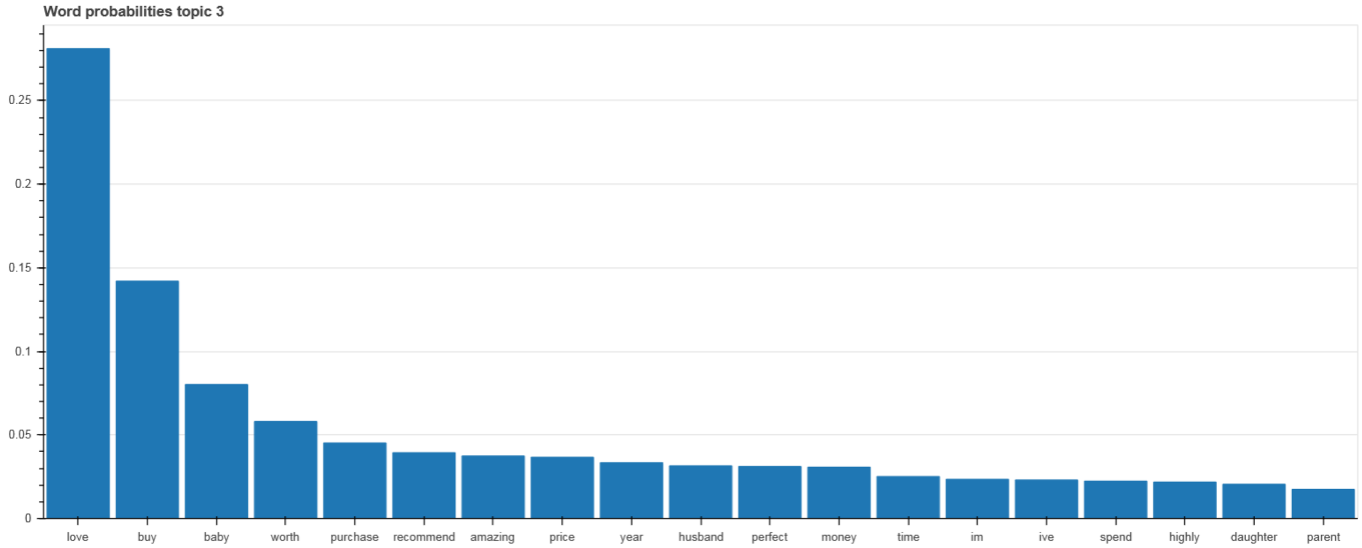


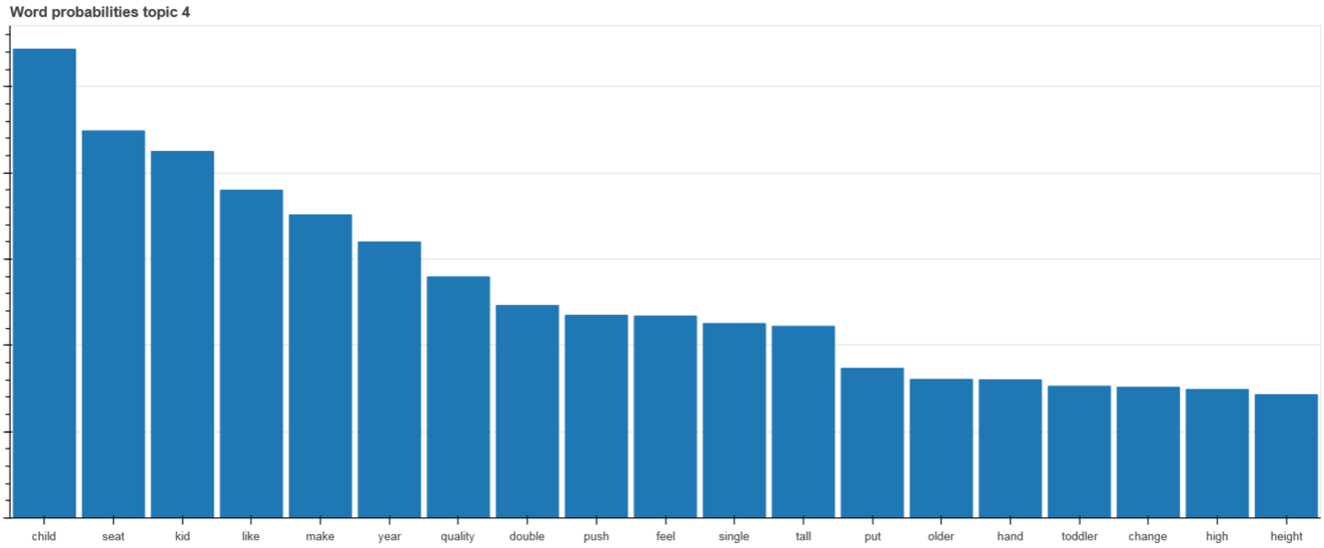
Word probabilities topic 0

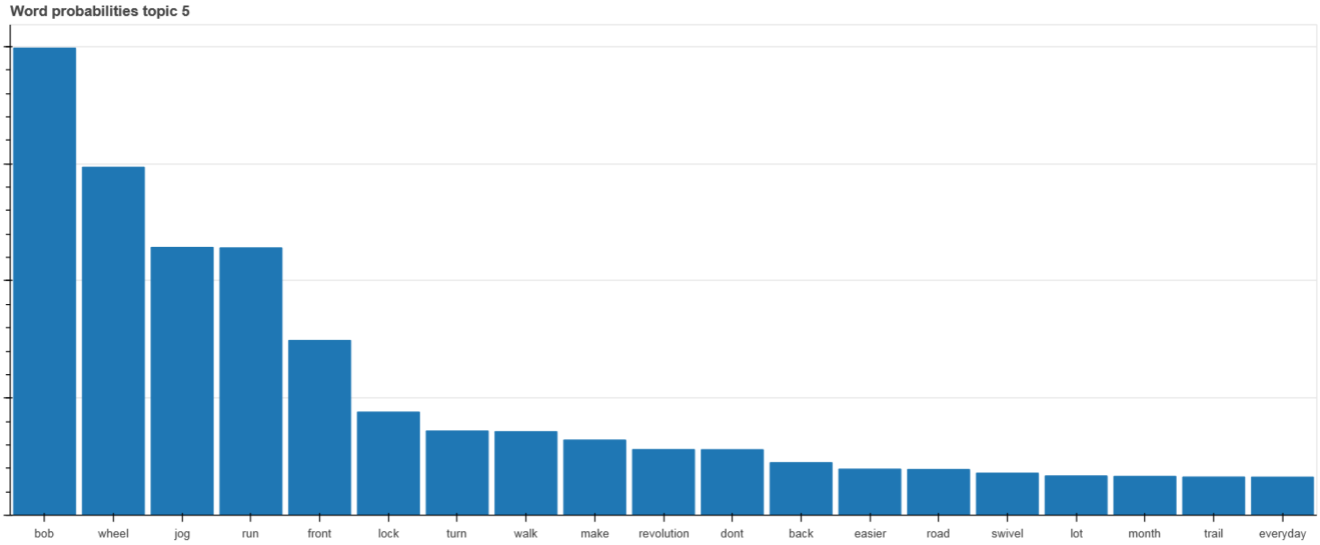


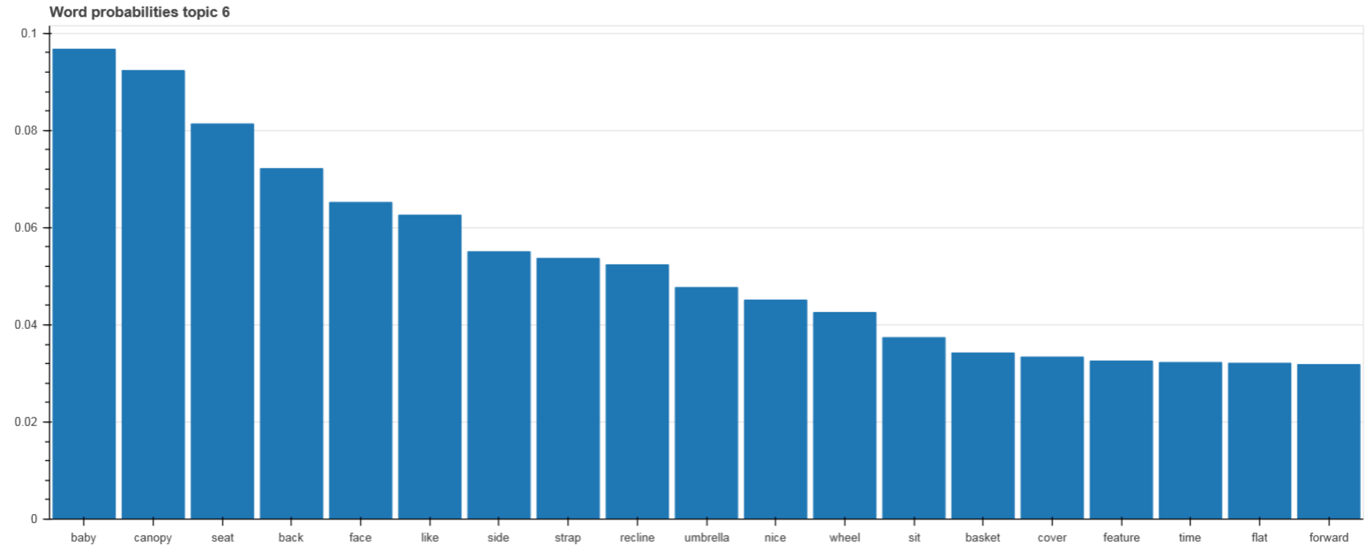
Word probabilities topic 1

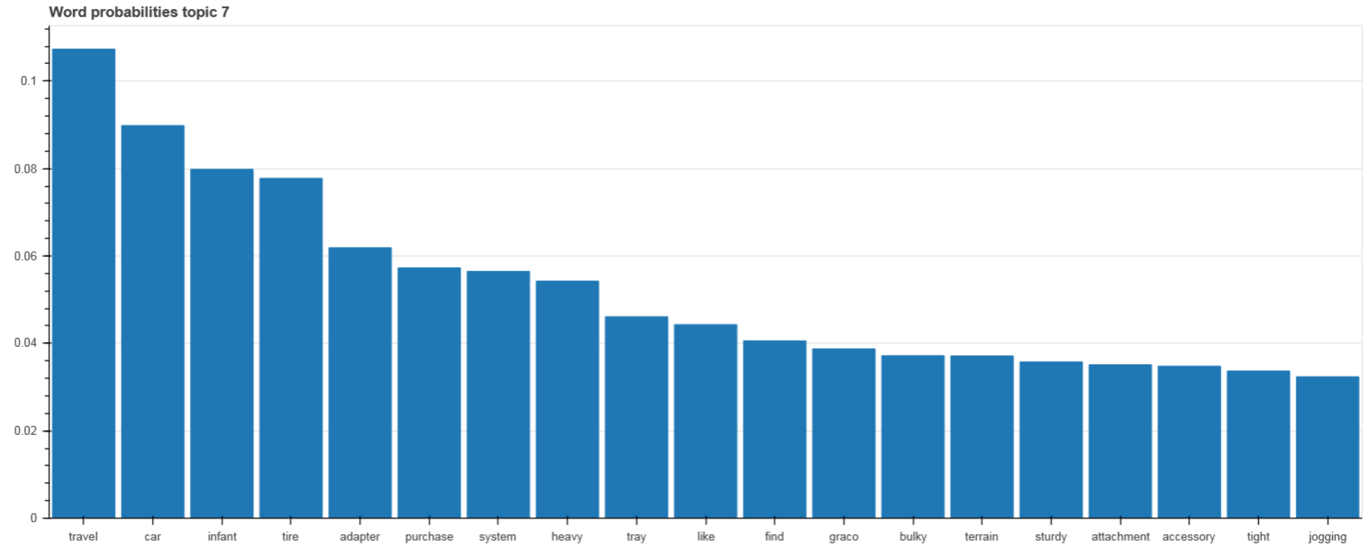

Word probabilities topic 2
0.12
0.1
0.08
0.06
0.04
0.02
0
baby
product
amazon
good
month
order
jogger
back
assemble
gift
break
receive
give
quality
time
customer
arrive
picture
find

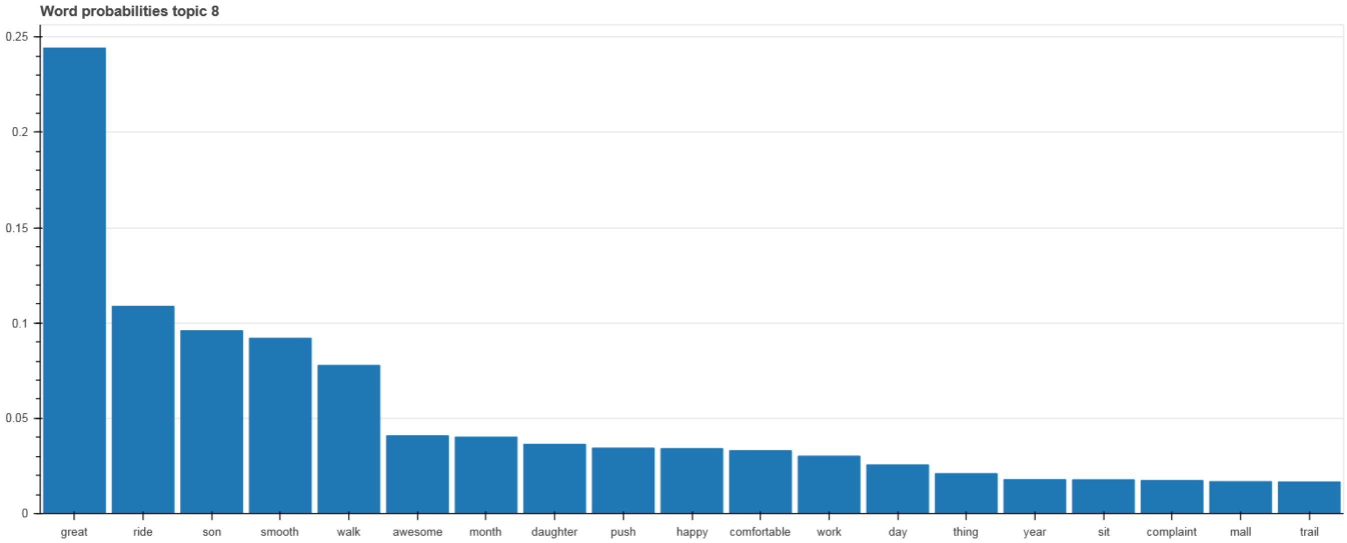

Word probabilities topic 3
0.25
0.2
0.15
0.1
0.05
0
love
buy
baby
worth
purchase
recommend
amazing
price
year
husband
perfect
money
time
im
ive
spend
highly
daughter
parent

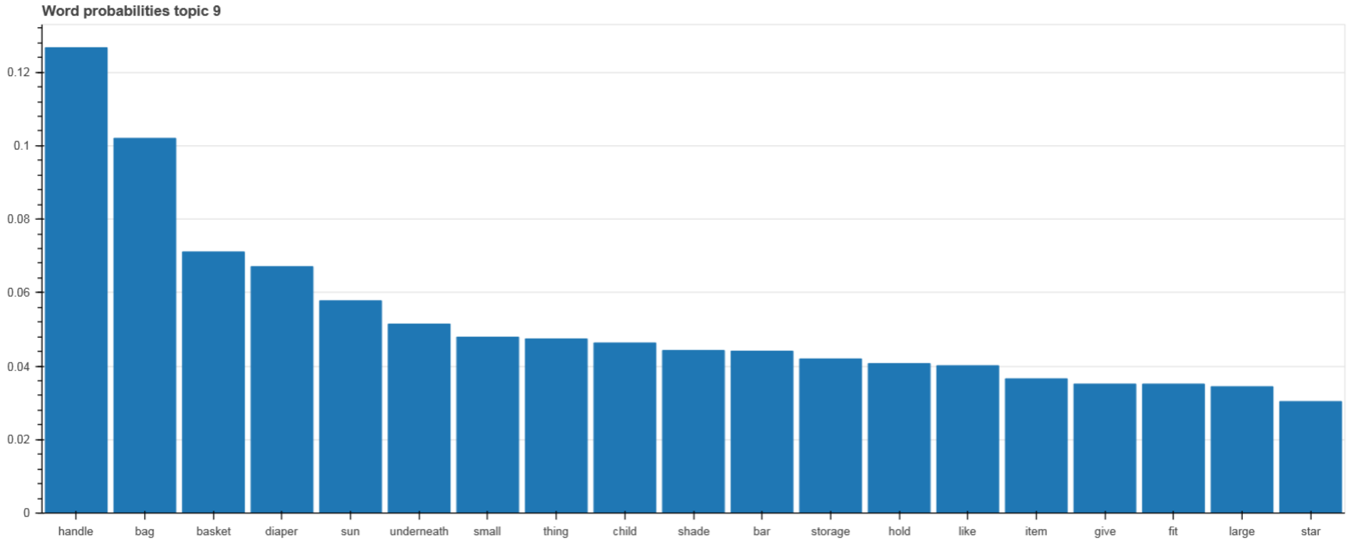

Word probabilities topic 4
0.1
0.08
0.06
0.04
0.02
0
child
seat
kid
like
make
year
quality
double
push
feel
single
tall
put
older
hand
toddler
change
high
height


Word probabilities topic 5
0.2
0.15
0.1
0.05
0
bob
wheel
jog
run
front
lock
turn
walk
make
revolution
dont
back
easier
road
swivel
lot
month
trail
everyday


Word probabilities topic 6
0.1
0.08
0.06
0.04
0.02
0
baby
canopy
seat
back
face
like
side
strap
recline
umbrella
nice
wheel
sit
basket
cover
feature
time
flat
forward


Word probabilities topic 7
0.1
0.08
0.06
0.04
0.02
0
travel
car
infant
tire
adapter
purchase
system
heavy
tray
like
find
graco
bulky
terrain
sturdy
attachment
accessory
tight
jogging


Word probabilities topic 8
0.25
0.2
0.15
0.1
0.05
0
great
ride
son
smooth
walk
awesome
month
daughter
push
happy
comfortable
work
day
thing
year
sit
complaint
mall
trail


Word probabilities topic 9
0.12
0.1
0.08
0.06
0.04
0.02
0
handle
bag
basket
diaper
sun
underneath
small
thing
child
shade
bar
storage
hold
like
item
give
fit
large
star

B.2. Top stroller reviews topic 5

Probability topic 5: $(\boldsymbol{\theta}_\mathbf{d})_5$	**Review**
0.921414981	Im an avid runner and a selfconfessed gearhead and was debating between this and the Ironman (or both) or some other brand. This was to be a pure running stroller, as we already have a good everyday stroller in the City Mini. It was a difficult choice, as we have some good local running trails, but theyre somewhat curvy. I wasnt sure if the Ironman would be too difficult to navigate through all the turns, and I thought maybe this one would be easier to run with the wheel unlocked. But I knew youre supposed to run with the wheel locked, and the Ironman was a bit backweighted to make it easier to turn so that made me think maybe the Ironman was the better choice. I got this one on basically a coin flip (after convincing myself that if backweighting is the big difference then I could just hang some weight on the handlebar) and it turned out that the one thing making me consider the Ironman probably makes no difference at all.So the thing with this stroller and curvy trails is, I still lock the wheel. Our trail (portage bicentennial) has some curvy areas and some straight areas. On the straight areas I found its far easier to run with the wheel locked, because you can guide the stroller with one hand and jog with the other. When you get to the curvy areas, you have to slow down a bit to turn it, but I dont think the Ironmans backweighting would make any difference in that respect. Because its not the lifting of the front wheel thats difficultits the angular momentum that youve got to account for to keep it from tipping over that forces you to slow down, and that would be applicable no matter which stroller youre using. The curvature at which I find myself slowing down is approx anything tighter than what a 1 4 mile track has. Wider than that (or anything under say 20 degrees no matter how sharp), its pretty easy to guide and turn with one hand and a couple flicks of the wrist. I cant imagine the Ironman being any easier.The advantage this has over the Ironman though, is that since we live in a condo on the 2nd floor, this is far easier to get out the door, through the elevator and hallways, etc, with the front wheel unlocked, and then lock the front wheel when we get outside. And even though weve got a walking stroller, sometimes thats in the trunk and the Bob is sitting out, and so its convenient to be able to take the Bob on walks (walks are easier with the wheel unlocked) in those situations. Also I like the tires better, as these are probably harder to puncture than the tires on the Ironman and have nicer tread. One possible disadvantage is that it seems like the front wheel needs recalibrated after every hour or so of run time. Not that it gets difficult to run with at that point at allitll be slightly noticeable on straightaways, but still well within the limits of what you can compensate for with just occasional flick of the wrist, and only takes 10 sec to bend over and recalibrate it. However I have no idea whether the Ironman would be any better in this regard with its permanently locked wheel or not, so this minor note may not even be relevant. And of course the Ironmans tires probably have a bit less drag, but Ill attest that the Revolutions tires drag is nexttonothing so if thats your concern, it shouldnt be.With respect to other strollers, I was also considering some that allow the wheel to be locked unlocked from a switch on the handlebar, thinking this might be a way to get through the curves on the trail without having to slow down. Now having the Bob, Id have to say I dont think that feature would make any difference, as to go from locked to unlocked and v.v., the front wheel has to do a 180 anyway, so youd have to slow down for that to happen. In fact to go from unlocked to locked, youd have to pull the stroller backwards to get the front wheel to lock facing forward. So, that realization combined with some generally negative reviews of strollers with that feature, Im glad I went with the Bob.
0.814486302	The front wheel locks on its own every time I turn and I cant fix it no matter what i do.
0.783527048	My granddaughter is now seventeen months old and I have been using this stroller for the past seven months. I bought this for jogging and hiking on dirt trails in our local parks, and for running in Central Park in Manhattan where my granddaughter lives. It does a superb job. However this is not a stroller that I would buy for everyday use, especially if you live in the suburbs rather than a big city, where people walk most places and dont need to constantly take a stroller in and out of a car trunk. Let me describe the stroller, which I bought at my local REI store since it came fully assembled and because they have a lifetime return policy in case I run into any problems.1. The construction is first rate, the fabric used is high quality, and it takes literally seconds to open and close, which is very simple. Just pull a red handle to lift the stroller into the open position. To close, just push two levers on the top forward and the stroller collapses (I disagree with the leading negative review that this is at all difficult). There is a wrist strap on the handle to be used when jogging that can be buckled to keep the stroller closed when it is folded. I NEVER use the wrist strap while jogging. I know that its purpose is to prevent the stroller from getting away from you if you lose your grip, but I think it is dangerous to use. If I tripped while jogging and had the wrist strap on, not only could I break my wrist from the force of the stroller with a child in it, but the odds are that I would flip the stroller too. If I felt myself going down I would rather just hold onto the handlebar and try to slow the stroller down. Just my opinion.2. The stroller has two modeswalk (the front wheel swivels) and jog (the front wheel is locked into position and stays straight). There is a simple red knob on the front wheel that allows you to easily switch between the two positions, and it literally takes just two seconds to switch.Note If you are ONLY going to jog, and dont mind having the front wheel permanently locked, then you can buy a less expensive BOB stroller known as the Sport Utility model, on which the front wheel does not swivel at all. Yes you can turn the stroller with a locked front wheelbut you have to lift the front of the stroller to do so. I did not want to be so limited, especially hiking on trails, which is why I bought the Revoution SE instead. However, after months of use I have found that lifting the front wheel when it is locked to change direction is not a big deal unless you are hiking in the woods on an uneven trail.This winter I went jogging with my then 13 month old granddaughter in Central Park in Manhattan and really appreciated how this stroller performed on lots of different surfaces and terrain smooth paved roads, uneven asphalt surfaces, sidewalks with bumps, street curbs, and some moderately steep uphill and downhill paths. (+300 words)

Probability topic 5: $(\boldsymbol{\theta}_\mathbf{d})_5$	**Review**
0.757429179	LOVE our Bob stroller. Dont know how we survived without it! Rarely lock the front wheel, and I use this for walking running. Its smooth, turns easily, and is a musthave for anyone who likes to exercise with their kiddo in tow.
0.751979336	If you are actually going to run with a stroller, this is the one you want. I had previously purchased a Baby Trend stroller and wish I would have paid more the first time for the Bob. The stroller runs so smoothly, especially with the front wheel locked in place. The other stroller wobbled when you ran and made it a lot more challenging to run.
0.736783563	My grandchild is one year old and I bought this stroller for jogging and hiking on dirt trails in our local parks. It does a superb job for each. However this is not a stroller that I would buy for everyday use, especially if you live in the suburbs rather than a big city, where people walk most places and dont need to constantly take a stroller in and out of a car trunk. Let me describe the stroller, which I bought at my local REI store since it came fully assembled and because they have a lifetime return policy in case I run into any problems.1. The construction is first rate, the fabric used is high quality, and it takes literally seconds to open and close, which is very simple. Just pull a red handle to lift the stroller into the open position. To close, just push two levers on the top forward and the stroller collapses (I disagree with the leading negative review that this is at all difficult). There is a wrist strap on the handle to be used when jogging that can be buckled to keep the stroller closed when it is folded. I NEVER use the wrist strap while jogging. I know that its purpose is to prevent the stroller from getting away from you if you lose your grip, but I think it is dangerous to use. If I tripped while jogging and had the wrist strap on, not only could I break my wrist from the force of the stroller with a child in it, but the odds are that I would flip the stroller too. If I felt myself going down I would rather just hold onto the handlebar and try to slow the stroller down. Just my opinion.2. The stroller has two modeswalk (the front wheel swivels) and jog (the front wheel is locked into position and stays straight). There is a simple red knob on the front wheel that allows you to easily switch between the two positions, and it literally takes just two seconds to switch.Note If you are ONLY going to jog, and dont mind having the front wheel permanently locked, then you can buy a less expensive BOB stroller known as the Sport Utility model, on which the front wheel does not swivel at all. Yes you can turn the stroller with a locked front wheelbut you have to lift the front of the stroller to do so. I did not want to be so limited, especially hiking on trails, which is why I bought the Revoution SE instead.3. This rides very smooth for jogging, and handles off road surface well when walking. The reason is that this is a very heavy stroller (24 poundsI weighed it on my luggage scale, which makes it heavier than any of the other strollers that my grandchild has, which I discuss below) and has very large wheelsagain larger than on her other strollers. Unlike other strollers, these wheels are inflatable just like bicycle tires. They need to be kept at 30psi for best performance. Of course jogging with a seriously under inflated wheel could be dangerous ordinary walking would just be more difficult. You dont have to check tire pressure all the time, but ask yourself if you want to bother having to check it at all if you are considering this as an everyday stroller. You might not want to have to deal with an unexpected flat tire just when you need to use the stroller.This takes up a lot of space in a trunk, and is heavy to put in and take out. Yes, each of the wheels has a quick release lever (just like bicycle wheels), so you can take them all off to save trunk space. This might make sense on a long trip, but I can tell you from experience that this is not something you would want to do on a regular basis, especially with a cranky young child or in inclement weather. Plus, using quick release wheels takes some getting used to. As the directions point out, if the quick release lever does not leave a visible imprint in the palm of your hand after you put the wheel back on, then you have not done it right.4. I do agree with the leading negative review that there is no soft padding on the seat, though I disagree that the crotch strap is too short (it is adjustable) or that buckling your child in is any more difficult than on any other stroller. When the canopy is fully extended, there is a window on top that lets you see your child. There is also ample storage underneath. You can adjust the seat to a reclining position using two straps, though for jogging you need to keep it fully upright (the further back it is, the less stability you have).However I would not use this as an everyday stroller. My grandchild (who lives in Manhattan) started out with the Bubaboo Cameleon stroller for local neighborhood walking (which I have reviewed on Amazon), and then at about nine months also started using the Maclaren Quest Sport stroller (which I have reviewed on Amazon) for traveling in cabs and subways, as well as day trips out of the city (like visiting me and my wife) since it is more light weight, easier to fold and close, and easier to carry with a carrying strap. And at my house she sometimes used the Graco Infant Car Seat stroller frame (which I have reviewed on Amazon).I mention these different strollers because all of these provide more comfortable seating, and are lighter and more compact (except maybe the heavier and bulkier Bugaboo) than the Revolution SE, which for me is a special purpose stroller for jogging and off road use. Yes, it can be used as an everyday stroller, but its strength lies not in lots of comfortable padding or a light weight compact size when folded, but rather in great stability while jogging or walking off road.For walking only the recommended age range is 8 weeks8months for jogging offroad use it is 8 months5 years. The stroller can accommodate a child up to 70 pounds.5. This stroller comes with a very clear and well illustrated manual that explains everything. Among the advanced features is a simple form of wheel alignment in case the stroller does not roll in a straight line (which could occur after off road use, the same as when a car goes over lots of bumps), and a shock absorber setting.Bottom line This is a special use stroller that works great for jogging and off road use. For everyday use I would get something else whether you live in the city or the suburbs.Update February 23, 2012 This past weekend I went jogging with my 13 month old granddaughter in Manhattan and really appreciated how this stroller performed on lots of different surfaces and terrain smooth paved roads, uneven asphalt surfaces, sidewalks with bumps, street curbs, and some moderately steep uphill and downhill paths. It was a breeze using this stroller and more importantly my granddaughter enjoyed every minute of it. Since it is critical to keep the front wheel in a locked position while running, anytime that I needed to make a turn (like at a street corner after we left Central Park), I easily just pulled back on the handlebars to lift the front wheel up and move it into the new position. Very easy to do and no big deal.Update June 10, 2012 The instruction manual contains the following warning in bold letters Never jog with the stroller in walk mode. Doing so could result in loss of control and serious injury. Nevertheless my daughter and soninlaw went running with my granddaughter in Central Park in Manhattan with the stroller in walk mode. My daughter said it worked fine, and made the stroller much easier to maneuver going back and forth to the Park and running inside the Park. I am not recommending this, but am simply pointing out someone elses experience... (+300 words)

B.3. Data sets

Table B.1: ‘Cats and dogs’ data set. Simple, small data set with distinctive topic clusters by construction.

Documents	**Preprocessed documents**
cats are animals	cats animals
dogs are canids	dogs canids
cats are fluffy	cats fluffy
dogs bark	dogs bark
cats meow	cats meow
fluffy are cats	fluffy cats
animals are large	animals large
dogs bite	dogs bite
cats scratch	cats scratch
dogs bite	dogs bite
cats scratch	cats scratch
dogs bark	dogs bark
cats are fluffy	cats fluffy
animals are cool	animals cool
not all animals are fluffy	animals fluffy
dogs are tough	dogs tough
canids are special	canids special
bark dogs	bark dogs
cool cats	cool cats

Table B.2: Simulated data for test of Adam optimization applied to determining the posterior mode estimates in the LDA with syntax and sentiment model. The data is simulated with hyperparameters $\boldsymbol{\alpha} = (0.5, 0.5)$, $\boldsymbol{\gamma} = (0.5, 0.5, 0.5)$, and $\boldsymbol{\beta_o}$ for $o = 1, 2, 3$ as explained in chapter 6. There are 20 documents in total, of which 11 are shown here. A period is used to indicate the end of a phrase.

Simulated data
people give give give give. donot donot donot donot donot. give walk walk give give walk. give people give give people fluffy. give give give give give. fluffy give give give give. give give people people give give give. give walk people give give. donot donot donot nice donot. donot people nice nice donot donot. nice donot donot donot donot donot donot. donot happiness donot donot donot donot
most noteworthy lifelong lifelong lifelong lifelong. afraid afraid smell sad smell afraid smell afraid. donot cat cat pet cat. cat pet cat cat pet cat. sad people afraid smell sad sad smell. most canid lifelong noteworthy noteworthy hate. lifelong noteworthy canid happiness most most. canid pet lifelong lifelong lifelong pet canid. noteworthy pet lifelong most pet most. cat pet smell pet cat
afraid smell smell sad sad. afraid people people sad smell people. afraid sad afraid afraid sad. afraid smell afraid smell afraid sad afraid. afraid smell afraid afraid afraid sad sad. sad sad people sad afraid. people afraid smell sad afraid. afraid afraid afraid smell afraid. smell afraid afraid afraid smell. afraid smell afraid afraid afraid afraid afraid. people afraid people sad afraid sad. smell smell afraid smell sad smell
lifelong most happiness canid noteworthy stubborn. pet canid lifelong hate lifelong most. noteworthy noteworthy lifelong lifelong most canid. canid most canid lifelong happiness lifelong hate. lifelong lifelong most pet most. lifelong canid lifelong lifelong hate. lifelong most lifelong happiness noteworthy noteworthy. canid pet canid pet lifelong canid. most noteworthy happiness lifelong cat stubborn pet. cat cat pet afraid make. smell cat pet pet cat. canid canid most hate most canid. hate most noteworthy most happiness
nice happiness people nice smell. give people walk walk walk. give give give give give give. give people give walk walk. smell pet pet give smell. people give give give give. give walk walk walk walk. people give walk give give. walk give give walk walk walk. cat cat people cat canid cat hate. give people walk give people. people walk walk people walk people. smell afraid smell afraid people smell afraid. cat wet cat cat cat cat pet
wet cat cat smell canid. canid cat cat cat afraid pet wet. canid hate lifelong noteworthy most lifelong hate. smell afraid afraid smell smell afraid smell. sad sad people afraid smell. canid most lifelong noteworthy lifelong pet. sad sad sad sad give. walk give give give people. lifelong hate noteworthy hate lifelong canid. cat cat like pet pet. happiness stubborn canid lifelong most. canid cat afraid canid cat cat cat. most most canid pet canid lifelong most
noteworthy lifelong noteworthy lifelong most lifelong most canid. smell afraid smell people afraid. afraid smell smell smell afraid smell afraid. smell smell smell sad people. sad sad smell sad smell smell. afraid sad sad smell afraid smell. afraid smell sad afraid smell sad. people smell afraid smell people smell. make give give pet nice. smell afraid afraid afraid smell smell
cat canid cat allergic cat smell afraid. donot sad afraid afraid smell. pet cat cat cat donot cat. cat donot wet canid donot. smell hate cat cat pet cat afraid. cat canid cat wet afraid wet cat pet. cat donot pet pet pet cat pet. cat cat cat make cat. canid cat donot cat cat cat. cat cat cat cat cat pet pet
most noteworthy happiness pet canid pet lifelong. hate hate noteworthy lifelong hate. canid most canid lifelong most canid lifelong. noteworthy lifelong canid most pet. lifelong noteworthy pet lifelong happiness noteworthy hate noteworthy. lifelong lifelong canid most noteworthy cat. lifelong lifelong happiness lifelong lifelong. lifelong most most stubborn canid canid most. noteworthy canid hate dog lifelong canid. lifelong noteworthy most lifelong hate hate. canid lifelong pet canid most cat lifelong canid. noteworthy pet happiness lifelong canid stubborn. most canid noteworthy noteworthy most canid hate hate
cat cat cat cat cat. give walk give walk give walk walk. cat like cat wet cat pet cat. lifelong lifelong canid lifelong canid canid canid pet. cat canid cat cat cat cat cat. allergic wet allergic like wet. hate canid most canid lifelong. cat cat pet cat cat cat cat pet. afraid cat pet cat cat cat afraid. walk give give give give. hate most lifelong lifelong lifelong lifelong
donot nice donot people donot happiness happiness nice happiness. nice donot nice donot donot nice. people nice donot donot donot. happiness nice donot donot donot people donot. happiness nice happiness happiness donot people nice. happiness nice nice happiness donot. donot donot donot happiness people. nice happiness donot nice nice. nice nice nice happiness donot happiness people. nice nice donot donot nice nice. happiness nice nice happiness nice donot. people donot donot nice donot happiness

Table B.3: Simulated data for test of Adam optimization applied to determining the posterior mode estimates in the LDA with syntax and sentiment model. The data is simulated with hyperparameters $\boldsymbol{\alpha} = (0.5, 0.5)$, $\boldsymbol{\gamma} = (0.5, 0.5, 0.5)$, and $\boldsymbol{\beta_o}$ for $o = 1, 2, 3$ as explained in chapter 6. There are 20 documents in total, of which 9 are shown here. A period is used to indicate the end of a phrase.

Simulated data
pet canid sad canid sad give smell. pet wet cat cat people pet. aggressive regret pet sad give pet. sad sad sad smell smell afraid. smell smell smell sad sad afraid people. smell smell smell smell sad sad. pet pet pet give pet nice. afraid sad afraid smell afraid. sad sad afraid afraid smell sad sad. walk people walk give give give give. give walk give walk give. afraid sad sad people smell afraid. donot donot nice people donot happiness donot
give give give give give walk. people canid make cat canid cat. people give give people give give. cat cat cat cat pet cat cat. pet afraid canid cat pet cat cat canid. give give give walk people people walk. make pet cat pet pet cat. pet cat cat cat cat canid. walk people people give walk give give. make pet canid pet cat. pet cat cat cat pet cat cat make. cat cat pet cat cat cat. cat hate canid cat pet cat pet cat. afraid cat cat smell pet pet pet. people give give give walk give
most pet hate noteworthy hate pet most. afraid smell afraid smell afraid smell. lifelong hate lifelong hate hate most. smell sad smell afraid afraid. lifelong stubborn noteworthy lifelong canid. smell smell smell sad sad sad donot. sad sad sad sad smell afraid. cat walk cat pet cat cat. cat afraid cat cat wet. most hate cat lifelong canid. canid hate noteworthy lifelong lifelong. canid cat canid cat cat
give give give give walk people people. walk cat people give give. give give give give give. give walk people people give people. people walk people give walk. fluffy give give give people. give people people walk people. walk give walk walk walk walk. people people people give give give. give fluffy people people give. people people give give walk. fluffy give walk give fluffy fluffy. give give walk give walk give
donot donot happiness donot happiness happiness nice nice donot. canid hate hate pet lifelong. lifelong most lifelong lifelong stubborn. lifelong canid lifelong lifelong dog pet pet. canid pet cat pet smell cat cat. cat noteworthy lifelong lifelong lifelong lifelong noteworthy stubborn. pet cat canid noteworthy lifelong noteworthy. lifelong most pet canid lifelong most. noteworthy hate cat pet canid wet most. noteworthy most canid most noteworthy most. lifelong lifelong canid lifelong canid noteworthy pet canid. noteworthy most canid canid lifelong hate happiness
pet sad nice pet like. noteworthy lifelong lifelong lifelong most canid most happiness. noteworthy hate lifelong hate canid most canid most. canid hate most canid noteworthy. donot afraid people afraid smell. pet sad pet smell give. most canid canid most most pet. smell afraid people smell sad sad sad afraid. sad smell smell afraid smell sad. hate canid most most hate most
hate lifelong canid canid lifelong hate. canid canid lifelong lifelong lifelong lifelong. lifelong most hate noteworthy noteworthy lifelong. noteworthy canid hate canid hate noteworthy. hate most pet noteworthy canid noteworthy. canid happiness happiness lifelong most lifelong canid. canid canid most noteworthy lifelong canid noteworthy most lifelong most happiness. lifelong noteworthy most canid canid. canid pet lifelong happiness lifelong. lifelong lifelong lifelong canid canid lifelong
smell smell smell afraid smell. sad smell pet make pet. give sad sad smell smell pet. make sad sad give pet sad pet. pet cat cat cat cat. give people give give give walk walk. give walk walk walk people. people give people give give. fluffy fluffy give walk give walk. sad make pet sad smell sad. give pet give nice sad like
sad sad smell smell people. sad afraid sad afraid afraid smell afraid. afraid afraid afraid afraid afraid smell. smell sad afraid people smell smell. smell smell smell people sad. afraid smell sad smell afraid afraid. afraid sad sad people smell smell. smell smell afraid smell afraid people people. donot donot nice nice donot donot nice donot nice. sad smell afraid sad smell smell afraid. afraid afraid afraid smell smell smell afraid

B.4. Conjunction word list

but
so
or
and
after
before
although
even though
because
as
if
as long as
provided that
till
until
unless
when
once
as soon as
while
whereas
in spite of
despite
in addition
furthermore
however
on the other hand
therefore
consequently
firstly
secondly
thirdly
finally
accordingly
also
anyway
besides
for example
for instance
further
hence
incidentally
indeed
in fact
instead
likewise
meanwhile
moreover
namely
of course
on the contrary
otherwise
nevertheless
nonetheless
similarly
so far
until now
then
therefore
thus

B.5. Stop word list

a
a’s
able
about
above
according
accordingly
across
actually
after
afterwards
again
against
ain’t
all
allow
allows
almost
alone
along
already
also
although
always
am
among
amongst
an
and
another
any
anybody
anyhow
anyone
anything
anyway
anyways
anywhere
apart
appear
appropriate
are
aren’t
around
as
aside
ask
asking
associated
at
available
away
awfully
b
be
became
because
become
becomes
becoming
been
before
beforehand
behind
being
believe
below
beside
besides
best
better
between
beyond
both
brief
but
by
c
c’mon
c’s
came
can
can’t
cannot
cant
certain
certainly
changes
clearly
co
com
come
comes
concerning
consequently
consider
considering
contain
containing
contains
corresponding
could
couldn’t
course
currently
d
definitely
described
despite
did
didn’t
different
do
does
doesn’t
doing
don’t
done
down
downwards
during
e
each
edu
eg
eight
either
else
elsewhere
enough
entirely
especially
et
etc
even
ever
every
everybody
everyone
everything
everywhere
ex
exactly
example
except
f
far
few
fifth
first
five
followed
following
follows
for
former
formerly
forth
four
from
further
furthermore
g
get
gets
getting
given
gives
go
goes
going
gone
got
gotten
greetings
h
had
hadn’t
happens
hardly
has
hasn’t
have
haven’t
having
he
he’s
hello
help
hence
her
here
here’s
hereafter
hereby
herein
hereupon
hers
herself
hi
him
himself
his
hither
hopefully
how
howbeit
however
i
i’d
i’ll
i’m
i’ve
ie
if
ignored
immediate
in
inasmuch
inc
indeed
indicate
indicated
indicates
inner
insofar
instead
into
inward
is
isn’t
it
it’d
it’ll
it’s
its
itself
j
just
k
keep
keeps
kept
know
knows
known
l
last
lately
later
latter
latterly
least
less
lest
let
let’s
likely
little
look
looking
looks
ltd
m
mainly
many
may
maybe
me
mean
meanwhile
merely
might
more
moreover
mostly
much
must
my
myself
n
name
namely
nd
near
nearly
necessary
need
needs
neither
never
nevertheless
new
next
nine
no
nobody
non
none
noone
nor
normally
not
nothing
novel
now
nowhere
o
obviously
of
off
often
oh
ok
okay
old
on
once
one
ones
only
onto
or
other

others
otherwise
ought
our
ours
ourselves
out
outside
over
overall
own
p
particular
particularly
per
perhaps
placed
please
plus
possible
presumably
probably
provides
q
que
quite
qv
r
rather
rd
re
really
reasonably
regarding
regardless
regards
relatively
respectively
right
s
said
same
saw
say
saying
says
second
secondly
see
seeing
seem
seemed
seeming
seems
seen
self
selves
sensible
sent
serious
seriously
seven
several
shall
she
should
shouldn’t
since
six
so
some
somebody
somehow
someone
something
sometime
sometimes
somewhat
somewhere
soon
specified
specify
specifying
still
sub
such
sup
sure
t
t’s
take
taken
tell
tends
th
than
thank
thanks
thanx
that
that’s
thats
the
their
theirs
them
themselves
then
thence
there
there’s
thereafter
thereby
therefore
therein
theres
thereupon
these
they
they’d
they’ll
they’re
they’ve
think
third
this
thorough
thoroughly
those
though
three
through
throughout
thru
thus
to
together
too
took
toward
towards
tried
tries
truly
try
trying
twice
two
u
un
under
unfortunately
unless
unlikely
until
unto
up
upon
us
use
used
uses
using
usually
uucp
v
value
various
very
via
viz
vs
w
want
wants
was
wasn’t
way
we
we’d
we’ll
we’re
we’ve
welcome
well
went
were
weren’t
what
what’s
whatever
when
whence
whenever
where
where’s
whereafter
whereas
whereby
wherein
whereupon
wherever
whether
which
while
whither
who
who’s
whoever
whole
whom
whose
why
will
willing
wish
with
within
without
won’t
wonder
would
would
wouldn’t
x
y
yes
yet
you
you’d
you’ll
you’re
you’ve
your
yours
yourself
yourselves
z
zero
you’re
you’ve
your
yours
yourself
yourselves
z
zero

B.6. Sentiment word lists

Positive words:

abidance
abide
abilities
ability
able
abound
above
above-average
absolve
abundance
abundant
accede
accept
acceptable
acceptance
accessible
acclaim
acclaimed
acclamation
accolade
accolades
accommodative
accomplish
accomplishment
accomplishments
accord
accordance
accordantly
accurate
accurately
achievable
achieve
achievement
achievements
acknowledge
acknowledgement
acquit
active
acumen
adaptability
adaptable
adaptive
adept
adeptly
adequate
adherence
adherent
adhesion
admirable
admirably
admiration
admire
admirer
admiring
admiringly
admission
admit
admittedly
adorable
adore
adored
adorer
adoring
adoringly
adroit
adroitly
adulate
adulation
adulatory
advanced
advantage
advantageous
advantages
adventure
adventuresome
adventurism
adventurous
advice
advisable
advocacy
advocate
affability
affable
affably
affection
affectionate
affinity
affirm
affirmation
affirmative
affluence
affluent
afford
affordable
afloat
agile
agilely
agility
agree
agreeability
agreeable
agreeableness
agreeably
agreement
allay
alleviate
allow
allowable
allure
alluring
alluringly
ally
almighty
altruist
altruistic
altruistically
amaze
amazed
amazement
amazing
amazingly
ambitious
ambitiously
ameliorate
amenable
amenity
amiability
amiabily
amiable
amicability
amicable
amicably
amity
amnesty
amour
ample
amply
amuse
amusement
amusing
amusingly
angel
angelic
animated
apostle
apotheosis
appeal
appealing
appease
applaud
appreciable
appreciate
appreciation
appreciative
appreciatively
appreciativeness
appropriate
approval
approve
apt
aptitude
aptly
ardent
ardently
ardor
aristocratic
arousal
arouse
arousing
arresting
articulate
ascendant
ascertainable
aspiration
aspirations
aspire
assent
assertions
assertive
asset
assiduous
assiduously
assuage
assurance
assurances
assure
assuredly
astonish
astonished
astonishing
astonishingly
astonishment
astound
astounded
astounding
astoundingly
astute
astutely
asylum
attain
attainable
attentive
attest
attraction
attractive
attractively
attune
auspicious
authentic
authoritative
autonomous
aver
avid
avidly
award
awe
awed
awesome
awesomely
awesomeness
awestruck
back
backbone
balanced
bargain
basic
bask
beacon
beatify
beauteous
beautiful
beautifully
beautify
beauty
befit
befitting
befriend
believable
beloved
benefactor
beneficent
beneficial
beneficially
beneficiary
benefit
benefits
benevolence
benevolent
benign
best
best-known
best-performing
best-selling
better
better-known
better-than-expected
blameless
bless
blessing
bliss
blissful
blissfully
blithe
bloom
blossom
boast
bold
boldly
boldness
bolster
bonny
bonus
boom
booming
boost
boundless
bountiful
brains
brainy
brave
bravery
breakthrough
breakthroughs
breathlessness
breathtaking
breathtakingly
bright
brighten
brightness
brilliance
brilliant
brilliantly
brisk
broad
brook
brotherly
bull
bullish
buoyant
calm
calming
calmness
candid
candor
capability
capable
capably
capitalize
captivate
captivating
captivation
care
carefree
careful
catalyst
catchy
celebrate
celebrated
celebration
celebratory
celebrity
champ
champion
charismatic
charitable
charity
charm
charming
charmingly
chaste
cheer
cheerful
cheery
cherish
cherished

cherub
chic
chivalrous
chivalry
chum
civil
civility
civilization
civilize
clarity
classic
clean
cleanliness
cleanse
clear
clear-cut
clearer
clearly
clever
closeness
clout
co-operation
coax
coddle
cogent
cohere
coherence
coherent
cohesion
cohesive
colorful
colossal
comeback
comely
comfort
comfortable
comfortably
comforting
commend
commendable
commendably
commensurate
commitment
commodious
commonsense
commonsensible
commonsensibly
commonsensical
compact
compassion
compassionate
compatible
compelling
compensate
competence
competency
competent
competitive
competitiveness
complement
compliant

compliment
complimentary
comprehensive
compromise
compromises
comrades
conceivable
conciliate
conciliatory
conclusive
concrete
concur
condone
conducive
confer
confidence
confident
confute
congenial
congratulate
congratulations
congratulatory
conquer
conscience
conscientious
consensus
consent
considerate
consistent
console
constancy
constructive
consummate
content
contentment
continuity
contribution
convenient
conveniently
conviction
convince
convincing
convincingly
cooperate
cooperation
cooperative
cooperatively
cordial
cornerstone
correct
correctly
cost-effective
cost-saving
courage
courageous
courageously
courageousness
court
courteous
courtesy
courtly

covenant
cozy
crave
craving
creative
credence
credible
crisp
crusade
crusader
cure-all
curious
curiously
cute
dance
dare
daring
daringly
darling
dashing
dauntless
dawn
daydream
daydreamer
dazzle
dazzled
dazzling
deal
dear
decency
decent
decisive
decisiveness
dedicated
defend
defender
defense
deference
definite
definitely
definitive
definitively
deflationary
deft
delectable
delicacy
delicate
delicious
delight
delighted
delightful
delightfully
delightfulness
democratic
demystify
dependable
deserve
deserved
deservedly
deserving
desirable

desire
desirous
destine
destined
destinies
destiny
determination
devote
devoted
devotee
devotion
devout
dexterity
dexterous
dexterously
dextrous
dig
dignified
dignify
dignity
diligence
diligent
diligently
diplomatic
discerning
discreet
discretion
discriminating
discriminatingly
distinct
distinction
distinctive
distinguish
distinguished
diversified
divine
divinely
dodge
dote
dotingly
doubtless
dream
dreamland
dreams
dreamy
drive
driven
durability
durable
dynamic
eager
eagerly
eagerness
earnest
earnestly
earnestness
ease
easier
easiest
easily
easiness

easy
easygoing
ebullience
ebullient
ebulliently
eclectic
economical
ecstasies
ecstasy
ecstatic
ecstatically
edify
educable
educated
educational
effective
effectiveness
effectual
efficacious
efficiency
efficient
effortless
effortlessly
effusion
effusive
effusively
effusiveness
egalitarian
elan
elate
elated
elatedly
elation
electrification
electrify
elegance
elegant
elegantly
elevate
elevated
eligible
elite
eloquence
eloquent
eloquently
emancipate
embellish
embolden
embrace
eminence
eminent
empower
empowerment
enable
enchant
enchanted
enchanting
enchantingly
encourage
encouragement
encouraging

encouragingly
endear
endearing
endorse
endorsement
endorser
endurable
endure
enduring
energetic
energize
engaging
engrossing
enhance
enhanced
enhancement
enjoy
enjoyable
enjoyably
enjoyment
enlighten
enlightenment
enliven
ennoble
enrapt
enrapture
enraptured
enrich
enrichment
ensure
enterprising
entertain
entertaining
enthral
enthrall
enthralled
enthuse
enthusiasm
enthusiast
enthusiastic
enthusiastically
entice
enticing
enticingly
entrance
entranced
entrancing
entreat
entreatingly
entrust
enviable
enviably
envision
envisions
epic
epitome
equality
equitable
erudite
especially
essential

established
esteem
eternity
ethical
eulogize
euphoria
euphoric
euphorically
even
evenly
eventful
everlasting
evident
evidently
evocative
exalt
exaltation
exalted
exaltedly
exalting
exaltingly
exceed
exceeding
exceedingly
excel
excellence
excellency
excellent
excellently
exceptional
exceptionally
excite
excited
excitedly
excitedness
excitement
exciting
excitingly
exclusive
excusable
excuse
exemplar
exemplary
exhaustive
exhaustively
exhilarate
exhilarating
exhilaratingly
exhilaration
exonerate
expansive
experienced
expert
expertly
explicit
explicitly
expressive
exquisite
exquisitely
extol
extoll
extraordinarily
extraordinary
exuberance
exuberant
exuberantly
exult
exultation
exultingly
fabulous
fabulously
facilitate
fair
fairly
fairness
faith
faithful
faithfully
faithfulness
fame
famed
famous
famously
fancy
fanfare
fantastic
fantastically
fantasy
farsighted
fascinate
fascinating
fascinatingly
fascination
fashionable
fashionably
fast-growing
fast-paced
fastest-growing
fathom
favor
favorable
favored
favorite
favour
fearless
fearlessly
feasible
feasibly
feat
featly
feisty
felicitate
felicitous
felicity
fertile
fervent
fervently
fervid
fervidly
fervor
festive
fidelity
fiery
fine
finely
first-class
first-rate
fit
fitting
flair
flame
flatter
flattering
flatteringly
flawless
flawlessly
flexible
flourish
flourishing
fluent
fond
fondly
fondness
foolproof
foremost
foresight
forgave
forgive
forgiven
forgiveness
forgiving
forgivingly
fortitude
fortuitous
fortuitously
fortunate
fortunately
fortune
fragrant
frank
free
freedom
freedoms
fresh
friend
friendliness
friendly
friends
friendship
frolic
fruitful
fulfillment
full-fledged
fun
functional
funny
gaiety
gaily
gain
gainful
gainfully
gallant
gallantly
galore
gem
gems
generosity
generous
generously
genial
genius
gentle
genuine
germane
giddy
gifted
glad
gladden
gladly
gladness
glamorous
glee
gleeful
gleefully
glimmer
glimmering
glisten
glistening
glitter
glorify
glorious
gloriously
glory
glossy
glow
glowing
glowingly
go-ahead
god-given
godlike
gold
golden
good
goodly
goodness
goodwill
gorgeous
gorgeously
grace
graceful
gracefully
gracious
graciously
graciousness
grail
grand
grandeur
grateful
gratefully
gratification
gratify
gratifying
gratifyingly
gratitude
great
greatest
greatness
greet
grin
grit
groove
groundbreaking
guarantee
guardian
guidance
guiltless
gumption
gush
gusto
gutsy
hail
halcyon
hale
hallowed
handily
handsome
handy
hanker
happily
happiness
happy
hard-working
hardier
hardy
harmless
harmonious
harmoniously
harmonize
harmony
haven
headway
heady
heal
healthful
healthy
heart
hearten
heartening
heartfelt
heartily
heartwarming
heaven
heavenly
help
helpful
herald
hero
heroic
heroically
heroine
heroize
heros
high-quality
highlight
hilarious
hilariously
hilariousness
hilarity
historic
holy
homage
honest
honestly
honesty
honeymoon
honor
honorable
hope
hopeful
hopefully
hopefulness
hopes
hospitable
hot
hug
humane
humanists
humanity
humankind
humble
humility
humor
humorous
humorously
humour
humourous
ideal
idealism
idealist
idealize
ideally
idol
idolize
idolized
idyllic
illuminate
illuminati
illuminating
illumine
illustrious
imaginative
immaculate
immaculately
impartial
impartiality
impartially
impassioned
impeccable
impeccably
impel
imperial
imperturbable
impervious
impetus
importance
important

importantly
impregnable
impress
impression
impressions
impressive
impressively
impressiveness
improve
improved
improvement
improving
improvise
inalienable
incisive
incisively
incisiveness
inclination
inclinations
inclined
inclusive
incontestable
incontrovertible
incorruptible
incredible
incredibly
indebted
indefatigable
indelible
indelibly
independence
independent
indescribable
indescribably
indestructible
indispensability
indispensable
indisputable
individuality
indomitable
indomitably
indubitable
indubitably
indulgence
indulgent
industrious
inestimable
inestimably
inexpensive
infallibility
infallible
infallibly
influential
informative
ingenious
ingeniously
ingenuity
ingenuous
ingenuously
ingratiate
ingratiating
ingratiatingly
innocence
innocent
innocently
innocuous
innovation
innovative
inoffensive
inquisitive
insight
insightful
insightfully
insist
insistence
insistent
insistently
inspiration
inspirational
inspire
inspiring
instructive
instrumental
intact
integral
integrity
intelligence
intelligent
intelligible
intercede
interest
interested
interesting
interests
intimacy
intimate
intricate
intrigue
intriguing
intriguingly
intuitive
invaluable
invaluablely
inventive
invigorate
invigorating
invincibility
invincible
inviolable
inviolate
invulnerable
irrefutable
irrefutably
irreproachable
irresistible
irresistibly
jauntily
jaunty
jest
joke
jollify
jolly
jovial
joy
joyful
joyfully
joyless
joyous
joyously
jubilant
jubilantly
jubilate
jubilation
judicious
just
justice
justifiable
justifiably
justification
justify
justly
keen
keenly
keenness
kemp
kid
kind
kindliness
kindly
kindness
kingmaker
kiss
knowledgeable
large
lark
laud
laudable
laudably
lavish
lavishly
law-abiding
lawful
lawfully
leading
lean
learned
learning
legendary
legitimacy
legitimate
legitimately
lenient
leniently
less-expensive
leverage
levity
liberal
liberalism
liberally
liberate
liberation
liberty
lifeblood
lifelong
light
light-hearted
lighten
likable
like
liking
lionhearted
literate
live
lively
lofty
logical
lovable
lovably
love
loveliness
lovely
lover
low-cost
low-risk
lower-priced
loyal
loyalty
lucid
lucidly
luck
luckier
luckiest
luckily
luckiness
lucky
lucrative
luminous
lush
luster
lustrous
luxuriant
luxuriate
luxurious
luxuriously
luxury
lyrical
magic
magical
magnanimous
magnanimously
magnetic
magnificence
magnificent
magnificently
magnify
majestic
majesty
manageable
manifest
manly
mannerly
marvel
marvellous
marvelous
marvelously
marvelousness
marvels
master
masterful
masterfully
masterpiece
masterpieces
masters
mastery
matchless
mature
maturely
maturity
maximize
meaningful
meek
mellow
memorable
memorialize
mend
mentor
merciful
mercifully
mercy
merit
meritorious
merrily
merriment
merriness
merry
mesmerize
mesmerizing
mesmerizingly
meticulous
meticulously
might
mightily
mighty
mild
mindful
minister
miracle
miracles
miraculous
miraculously
miraculousness
mirth
moderate
moderation
modern
modest
modesty
mollify
momentous
monumental
monumentally
moral
morality
moralize
motivate
motivated
motivation
moving
myriad
natural
naturally
navigable
neat
neatly
necessarily
necessary
neutralize
nice
nicely
nifty
nimble
noble
nobly
non-violence
non-violent
normal
notable
notably
noteworthy
noticeable
nourish
nourishing
nourishment
novel
nurture
nurturing
oasis
obedience
obedient
obediently
obey
objective
objectively
obliged
obviate
offbeat
offset
okay
onward
open
openly
openness
opportune
opportunity
optimal
optimism
optimistic
opulent
orderly
original
originality
outdo
outgoing
outshine
outsmart
outstanding

outstandingly
outstrip
outwit
ovation
overachiever
overjoyed
overture
pacifist
pacifists
painless
painlessly
painstaking
painstakingly
palatable
palatial
palliate
pamper
paradise
paramount
pardon
passion
passionate
passionately
patience
patient
patiently
patriot
patriotic
peace
peaceable
peaceful
peacefully
peacekeepers
peerless
penetrating
penitent
perceptive
perfect
perfection
perfectly
permissible
perseverance
persevere
persistent
personages
personality
perspicuous
perspicuously
persuade
persuasive
persuasively
pertinent
phenomenal
phenomenally
picturesque
piety
pillar
pinnacle
pious
pithy
placate
placid
plain
plainly
plausibility
plausible
playful
playfully
pleasant
pleasantly
please
pleased
pleasing
pleasingly
pleasurable
pleasurably
pleasure
pledge
pledges
plentiful
plenty
plush
poetic
poeticize
poignant
poise
poised
polished
polite
politeness
popular
popularity
portable
posh
positive
positively
positiveness
posterity
potent
potential
powerful
powerfully
practicable
practical
pragmatic
praise
praiseworthy
praising
pre-eminent
preach
preaching
precaution
precautions
precedent
precious
precise
precisely
precision
preeminent
preemptive
prefer
preferable
preferably
preference
preferences
premier
premium
prepared
preponderance
press
prestige
prestigious
prettily
pretty
priceless
pride
principle
principled
privilege
privileged
prize
pro
pro-American
pro-Beijing
pro-Cuba
pro-peace
proactive
prodigious
prodigiously
prodigy
productive
profess
proficient
proficiently
profit
profitable
profound
profoundly
profuse
profusely
profusion
progress
progressive
prolific
prominence
prominent
promise
promising
promoter
prompt
promptly
proper
properly
propitious
propitiously
prospect
prospects
prosper
prosperity
prosperous
protect
protection
protective
protector
proud
providence
prowess
prudence
prudent
prudently
punctual
pundits
pure
purification
purify
purity
purposeful
quaint
qualified
qualify
quasi-ally
quench
quicken
radiance
radiant
rally
rapport
rapprochement
rapt
rapture
raptureous
raptureously
rapturous
rapturously
rational
rationality
rave
re-conquest
readily
ready
reaffirm
reaffirmation
real
realist
realistic
realistically
reason
reasonable
reasonably
reasoned
reassurance
reassure
receptive
reclaim
recognition
recommend
recommendation
recommendations
recommended
recompense
reconcile
reconciliation
record-setting
recover
rectification
rectify
rectifying
redeem
redeeming
redemption
reestablish
refine
refined
refinement
reform
refresh
refreshing
refuge
regal
regally
regard
rehabilitate
rehabilitation
reinforce
reinforcement
rejoice
rejoicing
rejoicingly
relax
relaxed
relent
relevance
relevant
reliability
reliable
reliably
relief
relieve
relish
remarkable
remarkably
remedy
reminiscent
remunerate
renaissance
renewal
renovate
renovation
renown
renowned
repair
reparation
repay
repent
repentance
reputable
rescue
resilient
resolute
resolve
resolved
resound
resounding
resourceful
resourcefulness
respect
respectable
respectful
respectfully
respite
resplendent
responsibility
responsible
responsibly
responsive
restful
restoration
restore
restraint
resurgent
reunite
revel
revelation
revere
reverence
reverent
reverently
revitalize
revival
revive
revolution
reward
rewarding
rewardingly
rich
riches
richly
richness
right
righten
righteous
righteously
righteousness
rightful
rightfully
rightly
rightness
rights
ripe
risk-free
robust
romantic
romantically
romanticize
rosy
rousing
sacred
safe
safeguard
sagacity
sage
sagely
saint
saintliness
saintly
salable

salivate
salutary
salute
salvation
sanctify
sanction
sanctity
sanctuary
sane
sanguine
sanity
satisfaction
satisfactorily
satisfactory
satisfy
satisfying
savor
savvy
scenic
scruples
scrupulous
scrupulously
seamless
seasoned
secure
securely
security
seductive
selective
self-determination
self-respect
self-satisfaction
self-sufficiency
self-sufficient
semblance
sensation
sensational
sensationally
sensations
sense
sensible
sensibly
sensitive
sensitively
sensitivity
sentiment
sentimentality
sentimentally
sentiments
serene
serenity
settle
sexy
shelter
shield
shimmer
shimmering
shimmeringly
shine
shiny

shrewd
shrewdly
shrewdness
significance
significant
signify
simple
simplicity
simplified
simplify
sincere
sincerely
sincerity
skill
skilled
skillful
skillfully
sleek
slender
slim
smart
smarter
smartest
smartly
smile
smiling
smilingly
smitten
smooth
sociable
soft-spoken
soften
solace
solicitous
solicitously
solicitude
solid
solidarity
soothe
soothingly
sophisticated
sound
soundness
spacious
spare
sparing
sparingly
sparkle
sparkling
special
spectacular
spectacularly
speedy
spellbind
spellbinding
spellbindingly
spellbound
spirit
spirited
spiritual
splendid

splendidly
splendor
spotless
sprightly
spur
squarely
stability
stabilize
stable
stainless
stand
star
stars
stately
statuesque
staunch
staunchly
staunchness
steadfast
steadfastly
steadfastness
steadiness
steady
stellar
stellarly
stimulate
stimulating
stimulative
stirring
stirringly
stood
straight
straightforward
streamlined
stride
strides
striking
strikingly
striving
strong
studious
studiously
stunned
stunning
stunningly
stupendous
stupendously
sturdy
stylish
stylishly
suave
sublime
subscribe
substantial
substantially
substantive
subtle
succeed
success
successful
successfully

suffice
sufficient
sufficiently
suggest
suggestions
suit
suitable
sumptuous
sumptuously
sumptuousness
sunny
super
superb
superbly
superior
superlative
support
supporter
supportive
supreme
supremely
supurb
supurbly
sure
surely
surge
surging
surmise
surmount
surpass
survival
survive
survivor
sustainability
sustainable
sustained
sweeping
sweet
sweeten
sweetheart
sweetly
sweetness
swift
swiftness
sworn
tact
talent
talented
tantalize
tantalizing
tantalizingly
taste
temperance
temperate
tempt
tempting
temptingly
tenacious
tenaciously
tenacity
tender

tenderly
tenderness
terrific
terrifically
terrified
terrify
terrifying
terrifyingly
thank
thankful
thankfully
thinkable
thorough
thoughtful
thoughtfully
thoughtfulness
thrift
thrifty
thrill
thrilling
thrillingly
thrills
thrive
thriving
tickle
tidy
time-honored
timely
tingle
titillate
titillating
titillatingly
toast
togetherness
tolerable
tolerably
tolerance
tolerant
tolerantly
tolerate
toleration
top
torrid
torridly
tradition
traditional
tranquil
tranquility
treasure
treat
tremendous
tremendously
trendy
trepidation
tribute
trim
triumph
triumphal
triumphant
triumphantly
truculent

truculently
true
truly
trump
trumpet
trust
trusting
trustingly
trustworthiness
trustworthy
truth
truthful
truthfully
truthfulness
twinkly
ultimate
ultimately
ultra
unabashed
unabashedly
unanimous
unassailable
unbiased
unbosom
unbound
unbroken
uncommon
uncommonly
unconcerned
unconditional
unconventional
undaunted
understand
understandable
understanding
understate
understated
understatedly
understood
undisputable
undisputably
undisputed
undoubted
undoubtedly
unencumbered
unequivocal
unequivocally
unfazed
unfettered
unforgettable
uniform
uniformly
unique
unity
universal
unlimited
unparalleled
unpretentious
unquestionable
unquestionably
unrestricted

unscathed
unselfish
untouched
untrained
upbeat
upfront
upgrade
upheld
uphold
uplift
uplifting
upliftingly
upliftment
upright
upscale
upside
upward
urge
usable
useful
usefulness
utilitarian
utmost
uttermost
valiant
valiantly
valid
validity
valor
valuable
value
values
vanquish
vast
vastly
vastness
venerable
venerably
venerate
verifiable
veritable
versatile
versatility
viability
viable
vibrant
vibrantly
victorious
victory
vigilance
vigilant
vigorous
vigorously
vindicate
vintage
virtue
virtuous
virtuously
visionary
vital
vitality
vivacious
vivid
voluntarily
voluntary
vouch
vouchsafe
vow
vulnerable
want
warm
warmhearted
warmly
warmth
wealthy
welcome
welfare
well
well-being
well-connected
well-educated
well-established
well-informed
well-intentioned
well-managed
well-positioned
well-publicized
well-received
well-regarded
well-run
well-wishers
wellbeing
whimsical
white
wholeheartedly
wholesome
wide
wide-open
wide-ranging
will
willful
willfully
willing
willingness
wink
winnable
winners
wisdom
wise
wisely
wish
wishes
wishing
witty
wonder
wonderful
wonderfully
wonderous
wonderously
wondrous
woo
workable
world-famous
worship
worth
worth-while
worthiness
worthwhile
worthy
wow
wry
yearn
yearning
yearningly
yep
yes
youthful
zeal
zenith
zest

Negative words:

abandon
abandoned
abandonment
abase
abasement
abash
abate
abdicate
aberration
abhor
abhorred
abhorrence
abhorrent
abhorrently
abhors
abject
abjectly
abjure
abnormal
abolish
abominable
abominably
abominate
abomination
abrade
abrasive
abrupt
abscond
absence
absent-minded
absentee
absurd
absurdity
absurdly
absurdness
abuse
abuses
abusive
abysmal
abysmally
abyss
accidental
accost
accountable
accursed
accusation
accusations
accuse
accuses
accusing
accusingly
acerbate
acerbic
acerbically
ache
acrid
acridly
acridness
acrimonious
acrimoniously
acrimony
adamant
adamantly
addict
addiction
admonish
admonisher
admonishingly
admonishment
admonition
adrift
adulterate
adulterated
adulteration
adversarial
adversary
adverse
adversity
affectation
afflict
affliction
afflictive
affront
afraid
against
aggravate
aggravating
aggravation
aggression
aggressive
aggressiveness
aggressor
aggrieve
aggrieved
aghast
agitate
agitated
agitation
agitator
agonies
agonize
agonizing
agonizingly
agony
ail
ailment
aimless
airs
alarm
alarmed
alarming
alarmingly
alas
alienate
alienated
alienation
allegation
allegations
allege
allergic
aloof
altercation
although
ambiguity
ambiguous
ambivalence
ambivalent
ambush
amiss
amputate
anarchism
anarchist
anarchistic
anarchy
anemic
anger
angrily
angriness
angry
anguish
animosity
annihilate
annihilation
annoy
annoyance
annoyed
annoying
annoyingly
anomalous
anomaly
antagonism
antagonist
antagonistic
antagonize
anti-
anti-American
anti-Israeli
anti-Semites
anti-US
anti-occupation
anti-proliferation
anti-social
anti-white
antipathy
antiquated
antithetical
anxieties
anxiety
anxious
anxiously
anxiousness
apathetic
apathetically
apathy
ape
apocalypse
apocalyptic
apologist
apologists
appal
appall
appalled
appalling
appallingly
apprehension
apprehensions
apprehensive
apprehensively
arbitrary
arcane
archaic
arduous
arduously
argue
argument
argumentative
arguments
arrogance
arrogant
arrogantly
artificial
ashamed
asinine
asininely
asinininity
askance
asperse
aspersion
aspersions
assail
assassin
assassinate
assault
astray
asunder
atrocious
atrocities
atrocity
atrophy
attack
audacious
audaciously
audaciousness
audacity
austere
authoritarian
autocrat
autocratic
avalanche
avarice
avaricious
avariciously
avenge
averse
aversion
avoid
avoidance
awful
awfully
awfulness
awkward
awkwardness
ax
babble
backbite
backbiting
backward
backwardness
bad
badly
baffle
baffled
bafflement
baffling
bait
balk
banal
banalize
bane
banish
banishment
bankrupt
bar
barbarian
barbaric
barbarically
barbarity
barbarous
barbarously
barely
barren
baseless
bashful
bastard
battered
battering
battle
battle-lines
battlefield
battleground
batty
bearish
beast
beastly
bedlam
bedlamite
befoul
beg
beggar
beggarly
begging
beguile
belabor
belated
beleaguer
belie
belittle
belittled
belittling
bellicose
belligerence
belligerent
belligerently
bemoan
bemoaning
bemused
bent
berate
bereave
bereavement
bereft
berserk
beseech
beset
besiege
besmirch
bestial
betray
betrayal
betrayals
betrayer
bewail
beware
bewilder
bewildered
bewildering
bewilderingly
bewilderment
bewitch
bias
biased
biases
bicker
bickering
bid-rigging
bitch
bitchy
biting
bitingly
bitter
bitterly
bitterness
bizarre
blab
blabber
black
blackmail
blah

blame
blameworthy
bland
blandish
blaspheme
blasphemous
blasphemy
blast
blasted
blatant
blatantly
blather
bleak
bleakly
bleakness
bleed
blemish
blind
blinding
blindingly
blindness
blindside
blister
blistering
bloated
block
blockhead
blood
bloodshed
bloodthirsty
bloody
blow
blunder
blundering
blunders
blunt
blur
blurt
boast
boastful
boggle
bogus
boil
boiling
boisterous
bomb
bombard
bombardment
bombastic
bondage
bonkers
bore
boredom
boring
botch
bother
bothersome
bowdlerize
boycott
braggart
bragger
brainwash
brash
brashly
brashness
brat
bravado
brazen
brazenly
brazenness
breach
break
break-point
breakdown
brimstone
bristle
brittle
broke
broken-hearted
brood
browbeat
bruise
brusque
brutal
brutalising
brutalities
brutality
brutalize
brutalizing
brutally
brute
brutish
buckle
bug
bulky
bullies
bully
bullyingly
bum
bumpy
bungle
bungler
bunk
burden
burdensome
burdensomely
burn
busy
busybody
butcher
butchery
byzantine
cackle
cajole
calamities
calamitous
calamitously
calamity
callous
calumniate
calumniation
calumnies
calumnious
calumniously
calumny
cancer
cancerous
cannibal
cannibalize
capitulate
capricious
capriciously
capriciousness
capsize
captive
careless
carelessness
caricature
carnage
carp
cartoon
cartoonish
cash-strapped
castigate
casualty
cataclysm
cataclysmal
cataclysmic
cataclysmically
catastrophe
catastrophes
catastrophic
catastrophically
caustic
caustically
cautionary
cautious
cave
censure
chafe
chaff
chagrin
challenge
challenging
chaos
chaotic
charisma
chasten
chastise
chastisement
chatter
chatterbox
cheap
cheapen
cheat
cheater
cheerless
chide
childish
chill
chilly
chit
choke
choppy
chore
chronic
clamor
clamorous
clash
cliche
cliched
clique
clog
close
cloud
clumsy
coarse
cocky
coerce
coercion
coercive
cold
coldly
collapse
collide
collude
collusion
combative
comedy
comical
commiserate
commonplace
commotion
compel
complacent
complain
complaining
complaint
complaints
complex
complicate
complicated
complication
complicit
compulsion
compulsive
compulsory
concede
conceit
conceited
concern
concerned
concerns
concession
concessions
condemn
condemnable
condemnation
condescend
condescending
condescendingly
condescension
condolence
condolences
confess
confession
confessions
conflict
confound
confounded
confounding
confront
confrontation
confrontational
confuse
confused
confusing
confusion
congested
congestion
conspicuous
conspicuously
conspiracies
conspiracy
conspirator
conspiratorial
conspire
consternation
constrain
constraint
consume
contagious
contaminate
contamination
contempt
contemptible
contemptuous
contemptuously
contend
contention
contentious
contort
contortions
contradict
contradiction
contradictory
contrariness
contrary
contravene
contrive
contrived
controversial
controversy
convoluted
coping
corrode
corrosion
corrosive
corrupt
corruption
costly
counterproductive
coupists
covetous
cow
coward
cowardly
crackdown
crafty
cramped
cranky
crass
craven
cravenly
craze
crazily
craziness
crazy
credulous
crime
criminal
cringe
cripple
crippling
crisis
critic
critical
criticism
criticisms
criticize
critics
crook
crooked
cross
crowded
crude
cruel
cruelties
cruelty
crumble
crumple
crush
crushing
cry
culpable
cumbersome
cuplrit
curse
cursed
curses
cursory
curt
cuss
cut
cutthroat
cynical
cynicism
damage
damaging
damn
damnable
damnably
damnation
damned
damning
danger

dangerous
dangerousness
dangle
dark
darken
darkness
darn
dash
dastard
dastardly
daunt
daunting
dauntingly
dawdle
daze
dazed
dead
deadbeat
deadlock
deadly
deadweight
deaf
dearth
death
debacle
debase
debasement
debaser
debatable
debate
debauch
debaucher
debauchery
debilitate
debilitating
debility
decadence
decadent
decay
decayed
deceit
deceitful
deceitfully
deceitfulness
deceive
deceiver
deceivers
deceiving
deception
deceptive
deceptively
declaim
decline
declining
decrease
decreasing
decrement
decrepit
decrepitude
decry
deep
deepening
defamation
defamations
defamatory
defame
defeat
defect
defective
defensive
defiance
defiant
defiantly
deficiency
deficient
defile
defiler
deform
deformed
defrauding
defunct
defy
degenerate
degenerately
degeneration
degradation
degrade
degrading
degradingly
dehumanization
dehumanize
deign
deject
dejected
dejectedly
dejection
delinquency
delinquent
delirious
delirium
delude
deluded
deluge
delusion
delusional
delusions
demean
demeaning
demise
demolish
demolisher
demon
demonic
demonize
demoralize
demoralizing
demoralizingly
denial
denigrate
denounce
denunciate
denunciation
denunciations
deny
deplete
deplorable
deplorably
deplore
deploring
deploringly
deprave
depraved
depravedly
deprecate
depress
depressed
depressing
depressingly
depression
deprive
deprived
deride
derision
derisive
derisively
derisiveness
derogatory
desecrate
desert
desertion
desiccate
desiccated
desolate
desolately
desolation
despair
despairing
despairingly
desperate
desperately
desperation
despicable
despicably
despise
despised
despite
despoil
despoiler
despondence
despondency
despondent
despondently
despot
despotic
despotism
destabilisation
destitute
destitution
destroy
destroyer
destruction
destructive
desultory
deter
deteriorate
deteriorating
deterioration
deterrent
detest
detestable
detestably
detract
detraction
detriment
detrimental
devastate
devastated
devastating
devastatingly
devastation
deviate
deviation
devil
devilish
devilishly
devilment
devilry
devious
deviously
deviousness
devoid
diabolic
diabolical
diabolically
diametrically
diatribe
diatribes
dictator
dictatorial
differ
difficult
difficulties
difficulty
diffidence
dig
digress
dilapidated
dilemma
dilly-dally
dim
diminish
diminishing
din
dinky
dire
direly
direness
dirt
dirty
disable
disabled
disaccord
disadvantage
disadvantaged
disadvantageous
disaffect
disaffected
disaffirm
disagree
disagreeable
disagreeably
disagreement
disallow
disappoint
disappointed
disappointing
disappointingly
disappointment
disapprobation
disapproval
disapprove
disapproving
disarm
disarray
disaster
disastrous
disastrously
disavow
disavowal
disbelief
disbelieve
disbeliever
disclaim
discombobulate
discomfit
discomfititure
discomfort
discompose
disconcert
disconcerted
disconcerting
disconcertingly
disconsolate
disconsolately
disconsolation
discontent
discontented
discontentedly
discontinuity
discord
discordance
discordant
discountenance
discourage
discouragement
discouraging
discouragingly
discourteous
discourteously
discredit
discrepant
discriminate
discrimination
discriminatory
disdain
disdainful
disdainfully
disease
diseased
disfavor
disgrace
disgraced
disgraceful
disgracefully
disgruntle
disgruntled
disgust
disgusted
disgustedly
disgustful
disgustfully
disgusting
disgustingly
dishearten
disheartening
dishearteningly
dishonest
dishonestly
dishonesty
dishonor
dishonorable
dishonorablely
disillusion
disillusioned
disinclination
disinclined
disingenuous
disingenuously
disintegrate
disintegration
disinterest
disinterested
dislike
dislocated
disloyal
disloyalty
dismal
dismally
dismalness
dismay
dismayed
dismaying
dismayingly
dismissive
dismissively
disobedience
disobedient
disobey
disorder
disordered
disorderly
disorganized
disorient
disoriented
disown
disparage

disparaging
disparagingly
dispensable
dispirit
dispirited
dispiritedly
dispiriting
displace
displaced
displease
displeasing
displeasure
disproportionate
disprove
disputable
dispute
disputed
disquiet
disquieting
disquietingly
disquietude
disregard
disregardful
disreputable
disrepute
disrespect
disrespectable
disrespectablity
disrespectful
disrespectfully
disrespectfulness
disrespecting
disrupt
disruption
disruptive
dissatisfaction
dissatisfactory
dissatisfied
dissatisfy
dissatisfying
dissemble
dissembler
dissension
dissent
dissenter
dissention
disservice
dissidence
dissident
dissidents
dissocial
dissolute
dissolution
dissonance
dissonant
dissonantly
dissuade
dissuasive
distaste
distasteful
distastefully
distort
distortion
distract
distracting
distraction
distraught
distraughtly
distraughtness
distress
distressed
distressing
distressingly
distrust
distrustful
distrusting
disturb
disturbed
disturbed-let
disturbing
disturbingly
disunity
disvalue
divergent
divide
divided
division
divisive
divisively
divisiveness
divorce
divorced
dizzing
dizzingly
dizzy
doddering
dodgey
dogged
doggedly
dogmatic
doldrums
dominance
dominate
domination
domineer
domineering
doom
doomsday
dope
doubt
doubtful
doubtfully
doubts
down
downbeat
downcast
downer
downfall
downfallen
downgrade
downhearted
downheartedly
downside
drab
draconian
draconic
dragon
dragons
dragoon
drain
drama
drastic
drastically
dread
dreadful
dreadfully
dreadfulness
dreary
drones
droop
drought
drowning
drunk
drunkard
drunken
dubious
dubiously
dubitable
dud
dull
dullard
dumb
dumbfound
dumbfounded
dummy
dump
dunce
dungeon
dungeons
dupe
dusty
dwindle
dwindling
dying
earsplitting
eccentric
eccentricity
edgy
effigy
effrontery
ego
egocentric
egomania
egotism
egotistical
egotistically
egregious
egregiously
ejaculate
election-rigger
eliminate
elimination
emaciated
emasculate
embarrass
embarrassing
embarrassingly
embarrassment
embattled
embroil
embroiled
embroilment
emotional
empathize
empathy
emphatic
emphatically
emptiness
empty
encroach
encroachment
endanger
endless
enemies
enemy
enervate
enfeeble
enflame
engulf
enjoin
enmity
enormities
enormity
enormous
enormously
enrage
enraged
enslave
entangle
entanglement
entrap
entrapment
envious
enviously
enviousness
envy
epidemic
equivocal
eradicate
erase
erode
erosion
err
errant
erratic
erratically
erroneous
erroneously
error
escapade
eschew
esoteric
estranged
eternal
evade
evasion
evasive
evil
evildoer
evils
eviscerate
exacerbate
exacting
exaggerate
exaggeration
exasperate
exasperating
exasperatingly
exasperation
excessive
excessively
exclaim
exclude
exclusion
excoriate
excruciating
excruciatingly
excuse
excuses
execrate
exhaust
exhaustion
exhort
exile
exorbitant
exorbitantance
exorbitantly
expediencies
expedient
expel
expensive
expire
explode
exploit
exploitation
explosive
expose
exposed
expropriate
expropriation
expulse
expunge
exterminate
extermination
extinguish
extort
extortion
extraneous
extravagance
extravagant
extravagantly
extreme
extremely
extremism
extremist
extremists
fabricate
fabrication
facetious
facetiously
fading
fail
failing
failure
failures
faint
fainthearted
faithless
fake
fall
fallacies
fallacious
fallaciously
fallaciousness
fallacy
fallout
false
falsehood
falsely
falsify
famine
famished
fanatic
fanatical
fanatically
fanaticism
fanatics
fanciful
far-fetched
farce
farcical
farcical-yet-
provocat
ve farcically
farfetched
fascism
fascist
fastidious
fastidiously
fastuous
fat
fatal
fatalistic
fatalistically
fatally
fateful
fatefully
fathomless
fatigue
fatty
fatuity
fatuous
fatuously
fault
faulty
fawningly

faze
fear
fearful
fearfully
fears
fearsome
feckless
feeble
feeblely
feebleminded
feign
feint
fell
felon
felonious
ferocious
ferociously
ferocity
fetid
fever
feverish
fiasco
fiat
fib
fibber
fickle
fiction
fictional
fictitious
fidget
fidgety
fiend
fiendish
fierce
fight
figurehead
filth
filthy
finagle
fine
fissures
fist
flabbergast
flabbergasted
flagging
flagrant
flagrantly
flak
flake
flakey
flaky
flash
flashy
flat-out
flaunt
flaw
flawed
flaws
fleer
fleeting
flighty
flimflam
flimsy
flirt
flirty
floored
flounder
floundering
flout
fluster
foe
fool
foolhardy
foolish
foolishly
foolishness
forbid
forbidden
forbidding
force
forceful
foreboding
forebodingly
forfeit
forged
forget
forgetful
forgetfully
forgetfulness
forlorn
forlornly
formidable
forsake
forsaken
forswear
foul
foully
foulness
fractious
fractiously
fracture
fragile
fragmented
frail
frantic
frantically
franticly
fraternize
fraud
fraudulent
fraught
frazzle
frazzled
freak
freakish
freakishly
frenetic
frenetically
frenzied
frenzy
fret
fretful
friction
frictions
friggin
fright
frighten
frightening
frighteningly
frightful
frightfully
frigid
frivolous
frown
frozen
fruitless
fruitlessly
frustrate
frustrated
frustrating
frustratingly
frustration
fudge
fugitive
full-blown
fulminate
fumble
fume
fun
fundamentalism
furious
furiously
furor
fury
fuss
fussy
fustigate
fusty
futile
futilely
futility
fuzzy
gabble
gaff
gaffe
gaga
gaggle
gainsay
gainsayer
gall
galling
gallingly
gamble
game
gape
garbage
garish
gasp
gauche
gaudy
gawk
gawky
geezer
genocide
get-rich
ghastly
ghetto
gibber
gibberish
gibe
glare
glaring
glaringly
glib
glibly
glitch
gloatingly
gloom
gloomy
gloss
glower
glum
glut
gnawing
goad
goading
god-awful
goddam
goddamn
goof
gossip
graceless
gracelessly
graft
grandiose
grapple
grate
grating
gratuitous
gratuitously
grave
gravely
greed
greedy
grief
grievance
grievances
grieve
grieving
grievous
grievously
grill
grim
grimace
grind
gripe
grisly
gritty
gross
grossly
grotesque
grouch
grouchy
groundless
grouse
growl
grudge
grudges
grudging
grudgingly
gruesome
gruesomely
gruff
grumble
guile
guilt
guiltily
guilty
gullible
haggard
haggle
halfhearted
halfheartedly
hallucinate
hallucination
hamper
hamstring
hamstrung
handicapped
haphazard
hapless
harangue
harass
harassment
harboring
harbors
hard
hard-hit
hard-line
hard-liner
hardball
harden
hardened
hardheaded
hardhearted
hardliner
hardliners
hardly
hardship
hardships
harm
harmful
harms
harpy
harridan
harried
harrow
harsh
harshly
hassle
haste
hasty
hate
hateful
hatefully
hatefulness
hater
hatred
haughtily
haughty
haunt
haunting
havoc
hawkish
hazard
hazardous
hazy
headache
headaches
heartbreak
heartbreaker
heartbreaking
heartbreakingly
heartless
heartrending
heathen
heavily
heavy-handed
heavyhearted
heck
heckle
hectic
hedge
hedonistic
heedless
hegemonism
hegemonistic
hegemony
heinous
hell
hell-bent
hellion
helpless
helplessly
helplessness
heresy
heretic
heretical
hesitant
hideous
hideously
hideousness
hinder
hindrance
hoard
hoax
hobble
hole
hollow
hoodwink
hopeless
hopelessly
hopelessness
horde
horrendous
horrendously

horrible
horribly
horrid
horrific
horrifically
horrify
horrifying
horrifyingly
horror
horrors
hostage
hostile
hostilities
hostility
hotbeds
hothead
hotheaded
hothouse
hubris
huckster
humbling
humiliate
humiliating
humiliation
hunger
hungry
hurt
hurtful
hustler
hypocrisy
hypocrite
hypocrites
hypocritical
hypocritically
hysteria
hysteric
hysterical
hysterically
hysterics
icy
idiocies
idiocy
idiot
idiotic
idiotically
idiots
idle
ignoble
ignominious
ignominiously
ignominy
ignorance
ignorant
ignore
ill
ill-advised
ill-conceived
ill-fated
ill-favored
ill-mannered
ill-natured

ill-sorted
ill-tempered
ill-treated
ill-treatment
ill-usage
ill-used
illegal
illegally
illegitimate
illicit
illiquid
illiterate
illness
illogic
illogical
illogically
illusion
illusions
illusory
imaginary
imbalance
imbecile
imbroglio
immaterial
immature
imminence
imminent
imminently
immobilized
immoderate
immoderately
immodest
immoral
immorality
immorally
immovable
impair
impaired
impasse
impatience
impatient
impatiently
impeach
impedance
impede
impediment
impending
impenitent
imperfect
imperfectly
imperialist
imperil
imperious
imperiously
impermissible
impersonal
impertinent
impetuous
impetuously
impiety
impinge

impious
implacable
implausible
implausibly
implicate
implication
implode
impolite
impolitely
impolitic
importunate
importune
impose
imposers
imposing
imposition
impossible
impossiblity
impossibly
impotent
impoverish
impoverished
impractical
imprecate
imprecise
imprecisely
imprecision
improbability
improbable
improbably
improper
improperly
impropriety
imprudence
imprudent
impudence
impudent
impudently
impugn
impulsive
impulsively
impunity
impure
impurity
inability
inaccessible
inaccuracies
inaccuracy
inaccurate
inaccurately
inaction
inactive
inadequacy
inadequate
inadequately
inadverent
inadverently
inadvisable
inadvisably
inane
inanely

inappropriate
inappropriately
inapt
inaptitude
inarticulate
inattentive
incapable
incapably
incautious
incendiary
incense
incessant
incessantly
incite
incitement
incivility
inclement
incognizant
incoherence
incoherent
incoherently
incommensurate
incomparable
incomparably
incompatibility
incompatible
incompetence
incompetent
incompetently
incomplete
incompliant
incomprehensible
incomprehension
inconceivable
inconceivably
inconclusive
incongruous
incongruously
inconsequent
inconsequential
inconsequentially
inconsequently
inconsiderate
inconsiderately
inconsistence
inconsistencies
inconsistency
inconsistent
inconsolable
inconsolably
inconstant
inconvenience
inconvenient
inconveniently
incorrect
incorrectly
incorrigible
incorrigibly
incredulous
incredulously
inculcate

indecency
indecent
indecently
indecision
indecisive
indecisively
indecorum
indefensible
indefinite
indefinitely
indelicate
indeterminable
indeterminably
indeterminate
indifference
indifferent
indigent
indignant
indignantly
indignation
indignity
indiscernible
indiscreet
indiscreetly
indiscretion
indiscriminate
indiscriminately
indiscriminating
indisposed
indistinct
indistinctive
indoctrinate
indoctrination
indolent
indulge
ineffective
ineffectively
ineffectiveness
ineffectual
ineffectually
ineffectualness
inefficacious
inefficacy
inefficiency
inefficient
inefficiently
inelegance
inelegant
ineligible
ineloquent
ineloquently
inept
ineptitude
ineptly
inequalities
inequality
inequitable
inequitably
inequities
inertia
inescapable

inescapably
inessential
inevitable
inevitably
inexact
inexcusable
inexcusably
inexorable
inexorably
inexperience
inexperienced
inexpert
inexpertly
inexpiable
inexplainable
inexplicable
inextricable
inextricably
infamous
infamously
infamy
infected
inferior
inferiority
infernal
infest
infested
infidel
infidels
infiltrator
infiltrators
infirm
inflame
inflammatory
inflated
inflationary
inflexible
inflict
infraction
infringe
infringement
infringements
infuriate
infuriated
infuriating
infuriatingly
inglorious
ingrate
ingratitude
inhibit
inhibition
inhospitable
inhospitality
inhuman
inhumane
inhumanity
inimical
inimically
iniquitous
iniquity
injudicious

injure
injurious
injury
injustice
injustices
innuendo
inopportune
inordinate
inordinately
insane
insanely
insanity
insatiable
insecure
insecurity
insensible
insensitive
insensitively
insensitivity
insidious
insidiously
insignificance
insignificant
insignificantly
insincere
insincerely
insincerity
insinuate
insinuating
insinuation
insociable
insolence
insolent
insolently
insolvent
insouciance
instability
instable
instigate
instigator
instigators
insubordinate
insubstantial
insubstantially
insufferable
insufferably
insufficiency
insufficient
insufficiently
insular
insult
insulted
insulting
insultingly
insupportable
insupportably
insurmountable
insurmountably
insurrection
interfere
interference
intermittent
interrupt
interruption
intimidate
intimidating
intimidatingly
intimidation
intolerable
intolerablely
intolerance
intolerant
intoxicate
intractable
intransigence
intransigent
intrude
intrusion
intrusive
inundate
inundated
invader
invalid
invalidate
invalidity
invasive
invective
inveigle
invidious
invidiously
invidiousness
involuntarily
involuntary
irate
irately
ire
irk
irksome
ironic
ironies
irony
irrational
irrationality
irrationally
irreconcilable
irredeemable
irredeemably
irreformable
irregular
irregularity
irrelevance
irrelevant
irreparable
irreplacible
irrepressible
irresolute
irresolvable
irresponsible
irresponsibly
irretrievable
irreverence
irreverent
irreverently
irreversible
irritable
irritably
irritant
irritate
irritated
irritating
irritation
isolate
isolated
isolation
itch
jabber
jaded
jam
jar
jaundiced
jealous
jealously
jealousness
jealousy
jeer
jeering
jeeringly
jeers
jeopardize
jeopardy
jerk
jittery
jobless
joker
jolt
jumpy
junk
junky
juvenile
kaput
keen
kick
kill
killer
killjoy
knave
knife
knock
kook
kooky
lack
lackadaisical
lackey
lackeys
lacking
lackluster
laconic
lag
lambast
lambaste
lame
lame-duck
lament
lamentable
lamentably
languid
languish
languor
languorous
languorously
lanky
lapse
lascivious
last-ditch
laugh
laughable
laughably
laughingstock
laughter
lawbreaker
lawbreaking
lawless
lawlessness
lax
lazy
leak
leakage
leaky
least
lech
lecher
lecherous
lechery
lecture
leech
leer
leery
left-leaning
less
less-developed
lessen
lesser
lesser-known
letch
lethal
lethargic
lethargy
lewd
lewdly
lewdness
liability
liable
liar
liars
licentious
licentiously
licentiousness
lie
lier
lies
life-threatening
lifeless
limit
limitation
limited
limp
listless
litigious
little
little-known
livid
lividly
loath
loathe
loathing
loathly
loathsome
loathsomely
lone
loneliness
lonely
lonesome
long
longing
longingly
loophole
loopholes
loot
lorn
lose
loser
losing
loss
lost
lousy
loveless
lovelorn
low
low-rated
lowly
ludicrous
ludicrously
lugubrious
lukewarm
lull
lunatic
lunaticism
lurch
lure
lurid
lurk
lurking
lying
macabre
mad
madden
maddening
maddeningly
madder
madly
madman
madness
maladjusted
maladjustment
malady
malaise
malcontent
malcontented
maledict
malevolence
malevolent
malevolently
malice
malicious
maliciously
maliciousness
malign
malignant
malodorous
maltreatment
maneuver
mangle
mania
maniac
maniacal
manic
manipulate
manipulation
manipulative
manipulators
mar
marginal
marginally
martyrdom
martyrdom-seeking
massacre
massacres
maverick
mawkish
mawkishly
mawkishness
maxi-devaluation
meager
mean
meaningless
meanness
meddle
meddlesome
mediocre
mediocrity
melancholy
melodramatic
melodramatically
menace
menacing
menacingly
mendacious
mendacity
menial
merciless
mercilessly
mere
merely
mess
messy

midget
miff
militancy
mind
mindless
mindlessly
mirage
mire
misapprehend
misbecome
misbecoming
misbegotten
misbehave
misbehavior
miscalculate
miscalculation
mischief
mischievous
mischievously
misconception
misconceptions
miscreant
miscreants
misdirection
miser
miserable
miserableness
miserably
miseries
miserly
misery
misfit
misfortune
misgiving
misgivings
misguidance
misguide
misguided
mishandle
mishap
misinform
misinformed
misinterpret
misjudge
misjudgment
mislead
misleading
misleadingly
mislike
mismanage
misread
misreading
misrepresent
misrepresentation
miss
misstatement
mistake
mistakes
mistified
mistrust
mistrustful
mistrustfully
misunderstand
misunderstanding
misunderstandings
misunderstood
misuse
moan
mock
mockeries
mockery
mocking
mockingly
molest
molestation
monotonous
monotony
monster
monstrosities
monstrosity
monstrous
monstrously
moody
moon
moot
mope
morbid
morbidly
mordant
mordantly
moribund
mortification
mortified
mortify
mortifying
motionless
motley
mourn
mourner
mournful
mournfully
muddle
muddy
mudslinger
mudslinging
mulish
multi-polarization
mundane
murder
murderous
murderously
murky
muscle-flexing
mysterious
mysteriously
mystery
mystify
myth
nag
nagging
naive
naively
narrow
narrower
nastily
nastiness
nasty
nationalism
naughty
nauseate
nauseating
nauseatingly
nebulous
nebulously
need
needless
needlessly
needy
nefarious
nefariously
negate
negation
negative
neglect
neglected
negligence
negligent
negligible
nemesis
nervous
nervously
nervousness
nettle
nettlesome
neurotic
neurotically
niggle
nightmare
nightmarish
nightmarishly
nix
noisy
non-confidence
nonexistent
nonsense
nosey
notorious
notoriously
nuisance
numb
obese
object
objection
objectionable
objections
oblique
obliterate
obliterated
oblivious
obnoxious
obnoxiously
obscene
obscenely
obscenity
obscure
obscurity
obsess
obsession
obsessions
obsessive
obsessively
obsessiveness
obsolete
obstacle
obstinate
obstinately
obstruct
obstruction
obtrusive
obtuse
obviously
odd
odder
oddest
oddities
oddity
oddly
offence
offend
offending
offenses
offensive
offensively
offensiveness
officious
ominous
ominously
omission
omit
one-side
one-sided
onerous
onerously
onslaught
opinionated
opponent
opportunistic
oppose
opposition
oppositions
oppress
oppression
oppressive
oppressively
oppressiveness
oppressors
orphan
ostracize
outbreak
outburst
outbursts
outcast
outcry
outdated
outlaw
outmoded
outrage
outraged
outrageous
outrageously
outrageousness
outrages
outsider
over-acted
over-valuation
overact
overacted
overawe
overbalance
overbalanced
overbearing
overbearingly
overblown
overcome
overdo
overdone
overdue
overemphasize
overkill
overlook
overplay
overpower
overreach
overrun
overshadow
oversight
oversimplification
oversimplified
oversimplify
oversized
overstate
overstatement
overstatements
overtaxed
overthrow
overturn
overwhelm
overwhelming
overwhelmingly
overworked
overzealous
overzealously
pain
painful
painfully
pains
pale
paltry
pan
pandemonium
panic
panicky
paradoxical
paradoxically
paralize
paralyzed
paranoia
paranoid
parasite
pariah
parody
partiality
partisan
partisans
passe
passive
passiveness
pathetic
pathetically
patronize
paucity
pauper
paupers
payback
peculiar
peculiarly
pedantic
pedestrian
peeve
peeved
peevish
peevishly
penalize
penalty
perfidious
perfidity
perfunctory
peril
perilous
perilously
peripheral
perish
pernicious
perplex
perplexed
perplexing
perplexity
persecute
persecution
pertinacious
pertinaciously
pertinacity
perturb
perturbed
perverse
perversely
perversion
perversity
pervert
perverted
pessimism
pessimistic
pessimistically
pest
pestilent

petrified
petrify
pettifog
petty
phobia
phobic
phony
picky
pillage
pillory
pinch
pine
pique
pitiable
pitiful
pitifully
pitiless
pitilessly
pittance
pity
plagiarize
plague
plaything
plea
pleas
plebeian
plight
plot
plotters
ploy
plunder
plunderer
pointless
pointlessly
poison
poisonous
poisonously
polarisation
polemize
pollute
polluter
polluters
polution
pompous
poor
poorly
posturing
pout
poverty
powerless
prate
pratfall
prattle
precarious
precariously
precipitate
precipitous
predatory
predicament
prejudge
prejudice
prejudicial
premeditated
preoccupy
preposterous
preposterously
pressing
presume
presumptuous
presumptuously
pretence
pretend
pretense
pretentious
pretentiously
prevaricate
pricey
prickle
prickles
prideful
primitive
prison
prisoner
problem
problematic
problems
procrastinate
procrastination
profane
profanity
prohibit
prohibitive
prohibitively
propaganda
propagandize
proscription
proscriptions
prosecute
protest
protests
protracted
provocation
provocative
provoke
pry
pugnacious
pugnaciously
pugnacity
punch
punish
punishable
punitive
puny
puppet
puppets
puzzle
puzzled
puzzlement
puzzling
quack
qualms
quandary
quarrel
quarrellous
quarrellously
quarrels
quarrelsome
quash
queer
questionable
quibble
quit
quitter
racism
racist
racists
rack
radical
radicalization
radically
radicals
rage
ragged
raging
rail
rampage
rampant
ramshackle
rancor
rank
rankle
rant
ranting
rantingly
rascal
rash
rat
rationalize
rattle
ravage
raving
reactionary
rebellious
rebuff
rebuke
recalcitrant
recant
recession
recessionary
reckless
recklessly
recklessness
recoil
recourses
redundancy
redundant
refusal
refuse
refutation
refute
regress
regression
regressive
regret
regretful
regretfully
regrettable
regrettably
reject
rejection
relapse
relentless
relentlessly
relentlessness
reluctance
reluctant
reluctantly
remorse
remorseful
remorsefully
remorseless
remorselessly
remorselessness
renounce
renunciation
repel
repetitive
reprehensible
reprehensibly
reprehension
reprehensive
repress
repression
repressive
reprimand
reproach
reproachful
reprove
reprovingly
repudiate
repudiation
repugn
repugnance
repugnant
repugnantly
repulse
repulsed
repulsing
repulsive
repulsively
repulsiveness
resent
resentful
resentment
reservations
resigned
resistance
resistant
restless
restlessness
restrict
restricted
restriction
restrictive
retaliate
retaliatory
retard
reticent
retire
retract
retreat
revenge
revengeful
revengefully
revert
revile
reviled
revoke
revolt
revolting
revoltingly
revulsion
revulsive
rhapsodize
rhetoric
rhetorical
rid
ridicule
ridiculous
ridiculously
rife
rift
rifts
rigid
rigor
rigorous
rile
riled
risk
risky
rival
rivalry
roadblocks
rocky
rogue
rollercoaster
rot
rotten
rough
rubbish
rude
rue
ruffian
ruffle
ruin
ruinous
rumbling
rumor
rumors
rumours
rumple
run-down
runaway
rupture
rusty
ruthless
ruthlessly
ruthlessness
sabotage
sacrifice
sad
sadden
sadly
sadness
sag
salacious
sanctimonious
sap
sarcasm
sarcastic
sarcastically
sardonic
sardonically
sass
satirical
satirize
savage
savaged
savagely
savagery
savages
scandal
scandalize
scandalized
scandalous
scandalously
scandals
scant
scapegoat
scar
scarce
scarcely
scarcity
scare
scared
scarier
scariest
scarily
scarred
scars
scary
scathing
scathingly
scheme
scheming
scoff
scoffingly
scold
scolding
scoldingly
scorching
scorchingly
scorn
scornful
scornfully
scoundrel

scourge
scowl
scream
screech
screw
scum
scummy
second-class
second-tier
secretive
sedentary
seedy
seethe
seething
self-coup
self-criticism
self-defeating
self-destructive
self-humiliation
self-interest
self-interested
self-serving
selfinterested
selfish
selfishly
selfishness
senile
sensationalize
senseless
senselessly
serious
seriously
seriousness
sermonize
servitude
set-up
sever
severe
severely
severity
shabby
shadow
shadowy
shady
shake
shaky
shallow
sham
shambles
shame
shameful
shamefully
shamefulness
shameless
shamelessly
shamelessness
shark
sharp
sharply
shatter
sheer
shipwreck
shirk
shirker
shiver
shock
shocking
shockingly
shoddy
short-lived
shortage
shortchange
shortcoming
shortcomings
shortsighted
shortsightedness
showdown
shred
shrew
shriek
shrill
shrilly
shrivel
shroud
shrouded
shrug
shun
shunned
shy
shyly
shyness
sick
sicken
sickening
sickeningly
sickly
sickness
sidetrack
sidetracked
siege
sillily
silly
simmer
simplistic
simplistically
sin
sinful
sinfully
sinister
sinisterly
sinking
skeletons
skeptical
skeptically
skepticism
sketchy
skimpy
skittish
skittishly
skulk
slack
slander
slanderer
slanderous
slanderously
slanders
slap
slashing
slaughter
slaughtered
slaves
sleazy
slight
slightly
slime
sloppily
sloppy
sloth
slothful
slow
slow-moving
slowly
slug
sluggish
slump
slur
sly
smack
smash
smear
smelling
smokescreen
smolder
smoldering
smother
smoulder
smouldering
smug
smugly
smut
smuttier
smuttiest
smutty
snare
snarl
snatch
sneak
sneakily
sneaky
sneer
sneering
sneeringly
snub
so-cal
so-called
sob
sober
sobering
solemn
somber
sore
sorely
soreness
sorrow
sorrowful
sorrowfully
sorry
sounding
sour
sourly
spade
spank
spilling
spinster
spiritless
spite
spiteful
spitefully
spitefulness
split
splitting
spoil
spook
spookier
spookiest
spookily
spooky
spoon-fed
spoon-feed
spoonfed
sporadic
spot
spotty
spurious
spurn
sputter
squabble
squabbling
squander
squash
squirm
stab
stagger
staggering
staggeringly
stagnant
stagnate
stagnation
staid
stain
stake
stale
stalemate
stammer
stampede
standstill
stark
starkly
startle
startling
startlingly
starvation
starve
static
steal
stealing
steep
steeply
stench
stereotype
stereotypical
stereotypically
stern
stew
sticky
stiff
stifle
stifling
stiflingly
stigma
stigmatize
sting
stinging
stingingly
stink
stinking
stodgy
stole
stolen
stooge
stooges
storm
stormy
straggle
straggler
strain
strained
strange
strangely
stranger
strangest
strangle
strenuous
stress
stressful
stressfully
stricken
strict
strictly
strident
stridently
strife
strike
stringent
stringently
struck
struggle
strut
stubborn
stubbornly
stubbornness
stuffy
stumble
stump
stun
stunt
stunted
stupid
stupidity
stupidly
stupified
stupify
stupor
sty
subdued
subjected
subjection
subjugate
subjugation
submissive
subordinate
subservience
subservient
subside
substandard
subtract
subversion
subversive
subversively
subvert
succumb
sucker
suffer
sufferer
sufferers
suffering
suffocate
sugar-coat
sugar-coated
sugarcoated
suicidal
suicide
sulk
sullen
sully
sunder
superficial
superficiality
superficially
superfluous
superiority
superstition
superstitious
supposed
suppress
suppression
supremacy
surrender
susceptible
suspect
suspicion
suspicions
suspicious
suspiciously
swagger
swamped

swear
swindle
swipe
swoon
swore
sympathetic
sympathetically
sympathies
sympathize
sympathy
symptom
syndrome
taboo
taint
tainted
tamper
tangled
tantrum
tardy
tarnish
taunt
taunting
tauntingly
taunts
tawdry
taxing
tease
teasingly
tedious
tediously
temerity
temper
tempest
temptation
tense
tension
tentative
tentatively
tenuous
tenuously
tepid
terrible
terribleness
terribly
terror
terror-genic
terrorism
terrorize
thankless
thirst
thorny
thoughtless
thoughtlessly
thoughtlessness
thrash
threat
threaten
threatening
threats
throttle
throw
thumb
thumbs
thwart
timid
timidity
timidly
timidness
tiny
tire
tired
tiresome
tiring
tiringly
toil
toll
too
topple
torment
tormented
torrent
tortuous
torture
tortured
torturous
torturously
totalitarian
touchy
toughness
toxic
traduce
tragedy
tragic
tragically
traitor
traitorous
traitorously
tramp
trample
transgress
transgression
trauma
traumatic
traumatically
traumatize
traumatized
travesties
travesty
treacherous
treacherously
treachery
treason
treasonous
trial
trick
trickery
tricky
trivial
trivialize
trivially
trouble
troublemaker
troublesome
troublesomely
troubling
troublingly
truant
try
trying
tumultuous
turbulent
turmoil
twist
twisted
twists
tyrannical
tyrannically
tyranny
tyrant
ugh
ugliness
ugly
ulterior
ultimatum
ultimatums
ultra-hardline
unable
unacceptable
unacceptablely
unaccustomed
unattractive
unauthentic
unavailable
unavoidable
unavoidably
unbearable
unbearablely
unbelievable
unbelievably
uncertain
uncivil
uncivilized
unclean
unclear
uncollectible
uncomfortable
uncompetitive
uncompromising
uncompromisingly
unconfirmed
unconstitutional
uncontrolled
unconvincing
unconvincingly
uncouth
undecided
undefined
undependability
undependable
underdog
underestimate
underlings
undermine
underpaid
undesirable
undetermined
undid
undignified
undo
undocumented
undone
undue
unease
uneasily
uneasiness
uneasy
uneconomical
unequal
unethical
uneven
uneventful
unexpected
unexpectedly
unexplained
unfair
unfairly
unfaithful
unfaithfully
unfamiliar
unfavorable
unfeeling
unfinished
unfit
unforeseen
unfortunate
unfortunately
unfounded
unfriendly
unfulfilled
unfunded
ungovernable
ungrateful
unhappily
unhappiness
unhappy
unhealthy
unilateralism
unimaginable
unimaginably
unimportant
uninformed
uninsured
unipolar
unjust
unjustifiable
unjustifiably
unjustified
unjustly
unkind
unkindly
unlamentable
unlamentably
unlawful
unlawfully
unlawfulness
unleash
unlicensed
unlikely
unlucky
unmoved
unnatural
unnaturally
unnecessary
unneeded
unnerve
unnerved
unnerving
unnervingly
unnoticed
unobserved
unorthodox
unorthodoxy
unpleasant
unpleasantries
unpopular
unprecedent
unprecedented
unpredictable
unprepared
unproductive
unprofitable
unqualified
unravel
unraveled
unrealistic
unreasonable
unreasonably
unrelenting
unrelentingly
unreliability
unreliable
unresolved
unrest
unruly
unsafe
unsatisfactory
unsavory
unscrupulous
unscrupulously
unseemly
unsettle
unsettled
unsettling
unsettlingly
unskilled
unsophisticated
unsound
unspeakable
unspeakablely
unspecified
unstable
unsteadily
unsteadiness
unsteady
unsuccessful
unsuccessfully
unsupported
unsure
unsuspecting
unsustainable
untenable
untested
unthinkable
unthinkably
untimely
untrue
untrustworthy
untruthful
unusual
unusually
unwanted
unwarranted
unwelcome
unwieldy
unwilling
unwillingly
unwillingness
unwise
unwisely
unworkable
unworthy
unyielding
upbraid
upheaval
uprising
uproar
uproarious
uproariously
uproarous
uproarously
uproot
upset
upsetting
upsettingly
urgency
urgent
urgently
useless
usurp
usurper
utter
utterly
vagrant
vague
vagueness
vain
vainly
vanish
vanity
vehement
vehemently
vengeance
vengeful
vengefully
vengefulness
venom

venomous
venomously
vent
vestiges
veto
vex
vexation
vexing
vexingly
vice
vicious
viciously
viciousness
victimize
vie
vile
vileness
vilify
villainous
villainously
villains
villian
villianous
villianously
villify
vindictive
vindictively
vindictiveness
violate
violation
violator
violent
violently
viper
virulence
virulent
virulently
virus
vocally
vociferous
vociferously
void
volatile
volatility
vomit
vulgar
wail
wallow
wane
waning
wanton
war
war-like
warfare
warily
wariness
warlike
warning
warp
warped
wary
waste
wasteful
wastefulness
watchdog
wayward
weak
weaken
weakening
weakness
weaknesses
weariness
wearisome
weary
wedge
wee
weed
weep
weird
weirdly
whatever
wheedle
whimper
whine
whips
wicked
wickedly
wickedness
widespread
wild
wildly
wiles
wilt
wily
wince
withheld
withhold
woe
woebegone
woeful
woefully
worn
worried
worriedly
worrier
worries
worrisome
worry
worrying
worryingly
worse
worsen
worsening
worst
worthless
worthlessly
worthlessness
wound
wounds
wrangle
wrath
wreck
wrest
wrestle
wretch
wretched
wretchedly
wretchedness
writhe
wrong
wrongful
wrongly
wrought
yawn
yelp
zealot
zealous
zealously

Bibliography

[1] Matthew J Beal. *Variational algorithms for approximate Bayesian inference*. PhD thesis, University of London, 2003.

[2] JM Bernardo, MJ Bayarri, JO Berger, AP Dawid, D Heckerman, AFM Smith, M West, et al. The variational bayesian em algorithm for incomplete data: with application to scoring graphical model structures. *Bayesian statistics*, 7:453–464, 2003.

[3] Dimitri P Bertsekas and John N Tsitsiklis. Gradient Convergence In Gradient Methods With Errors. *SIAM Journal on Optimization*, 10(3):627–642, 2000.

[4] Bird, Steven and Klein, Ewan and Loper, Edward. *Natural Language Processing with Python*. O'Reilly Media, 2009.

[5] David M. Blei and Matthew D. Hoffman. Structured Stochastic Variational Inference. *Journal of Machine Learning Research*, 38:361–369, 2015. ISSN 00369543. doi: 10.1093/screen/38.3.282.

[6] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

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

[8] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[9] Thomas Carlyle. *Oliver Cromwell's Letter and Speeches*. Harper, 1855.

[10] William Darling and Fei Song. Probabilistic Topic and Syntax Modeling with Part-of-Speech LDA. *Arxiv*, pages 1–24, 2013.

[11] James M Dickey. Multiple Hypergeometric Functions: Probabilistic Interpretations and Statistical Uses. *Journal of the American Statistical Association*, 78(383):628–637, 2016.

[12] James M. Dickey, Jhy Ming Jiang, and Joseph B. Kadane. Bayesian methods for censored categorical data. *Journal of the American Statistical Association*, 82(399):773–781, 1987.

[13] John C Duchi. Introductory Lectures on Stochastic Optimization. Graduate Summer School Lectures, 2016.

[14] Bradley Efron and Trevor Hastie. *Computer age statistical inference: Algorithms, evidence, and data science*. Cambridge University Press, 2016. ISBN 9781316576533. doi: 10.1017/CBO9781316576533.

[15] Victoria Fromkin, Robert Rodman, and Nina Hyams. *An Introduction to Language*. Cengage Learning, 2011.

[16] Zoubin Ghahramani. Variational methods, part of course 'statistical approaches to learning and discovery', April 2003.

[17] Mohammad Shoaib Jameel. *Latent Probabilistic Topic Discovery for Text Documents Incorporating Segment Structure andWord Order*. PhD thesis, The Chinese University of Hong Kong (Hong Kong), 2014.

[18] Shoaib Jameel and Wai Lam. An unsupervised topic segmentation model incorporating word order. *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '13*, page 203, 2013.

[19] Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, pages 1–43, 2017.

[20] Thomas J. Jiang, Joseph B. Kadane, and James M. Dickey. Computation of Carlson's Multiple Hypergeometric Function R for Bayesian Applications. *Journal of Computational and Graphical Statistics*, 1(3): 231–251, 1992.

[21] Yohan Jo and Alice H Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 815–824. ACM, 2011.

[22] Diederik P. Kingma and Jimmy Lei Ba. ADAM: A method for stochastic optimization. In *ICLR conference proceedings*, 2015. ISBN 9780735412705. doi: 10.1063/1.4902458.

[23] KNIME AG. Knime. URL `https://www.knime.com`.

[24] Sergei Koltcov, Olessia Koltsova, and Sergey I. Nikolenko. Latent Dirichlet Allocation: Stability and Applications to Studies of User-generated Content. *Proceedings of the 2014 ACM Conference on Web Science*, pages 161–165, 2014.

[25] Fangtao Li, Minlie Huang, and Xiaoyan Zhu. Sentiment analysis with global topics and local dependency. In *AAAI*, volume 10, pages 1371–1376, 2010.

[26] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 375–384. ACM, 2009.

[27] Chenghua Lin, Yulan He, and Richard Everson. A comparative study of bayesian models for unsupervised sentiment detection. In *Proceedings of the fourteenth conference on computational natural language learning*, pages 144–152. Association for Computational Linguistics, 2010.

[28] Jun S. Liu. The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89(427):958–966, 1994.

[29] David G. Luenberger. *Optimization by vector space methods*. John Wiley & Sons, 1969.

[30] David J.C. MacKay. *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

[31] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*, pages 171–180. ACM, 2007.

[32] David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. Distributed algorithms for topic models. *Journal of Machine Learning Research*, 10(Aug):1801–1828, 2009.

[33] Kai Wang Ng, Guo-Liang Tian, and Man-Lai Tang. *Dirichlet and related distributions: Theory, methods and applications*, volume 888. John Wiley & Sons, 2011.

[34] John Paisley. A simple proof of the stick-breaking construction of the dirichlet process. Technical report, Princeton University, 2010.

[35] G. Parisi. *Statistical Field Theory*. Addison-Wesley, 1988.

[36] Michael Paul and Roxana Girju. A two-dimensional topic-aspect model for discovering multi-faceted topics. *Urbana*, 51(61801):36, 2010.

[37] Wiebe R. Pestman. *Mathematical Statistics*. Walter de Gruyter GmbH & Co, 1998.

[38] Rajesh Ranganath, Dustin Tran, and David Blei. Hierarchical variational models. In *International Conference on Machine Learning*, pages 324–333, 2016.

[39] Sebastian Ruder. An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747, 2016.

[40] J. Sethuraman. A constructive definition of dirichlet priors. *Statistics Sinica*, 4:639–650, 1994.

[41] A.F.M Smith and G.O. Roberts. Bayesian Computation Via the Gibbs sampler and Related Markov Chain Monte Carlo Methods. *Journal of the Royal Statistical Society*, 55(1):1–24, 1993.

[42] A.F.M Smith and G.O. Roberts. Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms. *Stochastic Processes and Their Applications*, 49:207–216, 1994.

[43] Mark Steyvers and Tom Griffiths. *Probabilistic Topic Models*, volume 3. 2007.

[44] Ivan Titov and Ryan McDonald. A joint model of text and aspect ratings for sentiment summarization. *proceedings of ACL-08: HLT*, pages 308–316, 2008.

[45] Viet Hung Tran. Copula Variational Bayes inference via information geometry. *IEEE Transactions on Information Theory*, pages 1–23, 2018.

[46] Frank van der Meulen. Lecture notes statistical inference (wi4455), December 2017.

[47] Martin J. Wainwright and Michael I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2007.

[48] Hanna M Wallach and David Mimno. Evaluation Methods for Topic Models. In *Proceedings of the 26th International Conference on Machine Learning*, pages 1105–1112, 2009.

[49] Hanna M. Wallach, David M. Mimno, and Andrew McCallum. *Rethinking LDA: Why Priors Matter.* Curran Associates, Inc., 2009.

[50] Hongning Wang, Yue Lu, and Chengxiang Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 783–792. ACM, 2010.

[51] Xuerui Wang and Andrew McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433. ACM, 2006.

[52] Limin Yao, David Mimno, and Andrew McCallum. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 937–946. ACM, 2009.

[53] Cheng Zhang, Judith Butepage, Hedvig Kjellstrom, and Stephan Mandt. Advances in Variational Inference. *arXiv preprint arXiv:1711.05597*, 2017.