Hierarchical Multiclass Topic Modelling with Prior Knowledge

Master's Thesis submitted to

First Supervisor: **Prof. Dr. Wolfgang K. Härdle**

Second Supervisor: **Prof. Dr. Cathy Yi-Hsuan Chen**

Ladislaus von Bortkiewicz Chair of Statistics

C.A.S.E. – Center for Applied Statistics

and Economics

Humboldt–Universität zu Berlin

by

Ken Schröder

(568399)

in partial fulfillment of the requirements

for the degree of

Master of Science (Statistics)

Berlin, January 29th, 2018

Abstract

A new multi-label document classification technique called CascadeLDA is introduced in this thesis. Rather than focusing on discriminative modelling techniques, CascadeLDA extends a baseline generative model by incorporating two types of prior information. Firstly, knowledge from a labeled training dataset is used to direct the generative model. Secondly, the implicit tree structure of the labels is exploited to emphasise discriminative features between closely related labels. By segregating the classification problem in an ensemble of smaller problems, out-of-sample results are achieved at about 25 times the speed of the baseline model. In this thesis, CascadeLDA is performed on datasets with academic abstracts and full academic papers. The model is employed to assist authors in tagging their newly published articles.

Keywords: Bayesian, Gibbs sampling, Latent Dirichlet Allocation, machine learning, natural language processing, topic modelling, variational inference.

Contents

List of Figures

List of Tables

1 Introduction

Automated text classification has been an active field of research for decades and is used in numerous applications. Despite its age, the field has gained prominence over the past decade. This is due to the increasing need to organise vast amounts of digital textual information and developments in machine learning techniques and computational power.

A number of equally valid interpretations of "automated text classification" prevail (Sebastiani, 2002). The interpretations range from solely identifying categories in a body of documents to tagging documents according to pre-specified categories. In this thesis, we will focus on the latter interpretation. An extensive variety of methods exist to tag documents. Most of them use a dataset of labeled texts with which a model is trained to predict the category (categories) to which the text belongs. This training may involve neural networks (Nam et al., 2014), support vector machines (Joachims, 2002) or simpler methods like naive Bayes (McCallum and Nigam, 1998) to name a few.

In this research, Latent Dirichlet Allocation (LDA) (Blei et al., 2003) will be used as a baseline method. LDA is a "generative probabilistic model for collections of discrete data such as text corpora" (Blei et al., 2003, p. 993) and is an unsupervised machine learning algorithm. A common application of LDA is to identify latent semantic topics contained in documents. However, many authors have proposed extensions to serve other purposes such as document classification (e.g. Blei and McAuliffe (2007); Ramage et al. (2009); Rubin et al. (2012)) and targeted topic identification (e.g. Jagarlamudi et al. (2012); Wang et al. (2016)). It is often the availability of prior information on the corpus that triggers researchers to incorporate additional features in the baseline LDA framework.

A representative corpus of academic economic papers and abstracts is utilised for this paper. Each document in the dataset is assigned to one or more labels that correspond to the academic discipline of the paper. *The American Economic Association* (AEA) introduced a tagging system for their academic articles which is now the standard in economic literature: JEL (*Journal of Economic Literature*) codes. The presence and nature of the JEL codes constitute the main source of prior information and form the basis for analysis.

The aim of this research is to develop an LDA-based document classification model that assists authors in accurately tagging their academic publications.

In Section 2 the general framework for Latent Dirichlet Allocation will be thoroughly dealt with, including elaborate background theory of conjugate priors, (collapsed) Gibbs sampling, exponential families and variational inference. Section 3 will delve into the dataset

at hand and how the prior information can be integrated in LDA. This section will introduce and discuss two existing LDA flavours, called Labeled LDA (L-LDA) and Hierarchically Supervised LDA (HSLDA), and formulate a new LDA extension by the name of CascadeLDA. This is followed by an introduction and critical assessment of the quality and characteristics of the datasets used. Section 5 is used to point out challenges in the classification problem, introduce evaluation metrics and discuss the variable settings used in the final models. The outcome of the models in the different datasets are presented in Section 6. This section also provides a detailed discussion to emphasise the fundamental differences between the three LDA extensions. Finally, Section 7 gives an overview of the findings, use-cases and suggestions for future research.

2 Latent Dirichlet Allocation

In order to stay within the scope of this paper, this section will only discuss Latent Dirichlet Allocation in the context of text modelling. Applications to other discrete data structures such as image and audio classification will therefore be ignored here. This section will not only provide outcomes and ready-to-use expressions, it will also deal with the mechanics and properties of the underlying distributions in detail. These insights are required during the development and analysis of the LDA extensions. Additionally, much attention will be given to collapsed Gibbs sampling and variational inference.

LDA views documents in a corpus as bags-of-words: Sentence structures are ignored completely. Every document is assumed to be a mixture of K topics. Each topic is assumed to be a categorical distribution over all words in the corpus. LDA is a fully generative model that assumes that a document's words are the result of a mixture of topics, from which words are drawn.

The only observable in the fundamental model are the documents and therefore their words. The aim of LDA is to identify distributions that represent (latent) semantic topics in the corpus and represent documents as a (latent) mixture of these topics. These two distributions are referred to as the *topic-word distributions* and *document-topic distributions*, respectively. The topic-word distribution of topic k follows a $\text{Categorical}_V(\boldsymbol{\phi}_k)$ distribution, and document d's document-topic distribution follows a $\text{Categorical}_K(\boldsymbol{\theta}_d)$ distribution.

The graphical model representing LDA can be found in Figure 1. The conditional relations

Variable	Values	Meaning
K	$\mathbb{N}_+$	Number of topics
D	$\mathbb{N}_+$	Number of documents
V	$\mathbb{N}_+$	Number of unique words in vocabulary
k, d, v	$\mathbb{N}_+$	Iterators indicating topic k, document d and word key v. $k = 1 \ldots, K$ and $d = 1, \ldots, D$ and $v = 1, \ldots, V$
N_d	$\mathbb{N}_+$	Number of words in document d
N	$\mathbb{N}_+$	Total number of words in all documents: $N = \sum_{d=1}^{D} N_d$
α	$\mathbb{R}_+$	Uninformative prior weight. Same for all d and k
$\alpha_{d,k}$	$\mathbb{R}_+$	Prior weight of topic k in document d
$\boldsymbol{\alpha}_d$	$\mathbb{R}_+^K$	$= (\alpha_{d,1}, \alpha_{d,2}, \ldots, \alpha_{d,K}) = \alpha_{d,1:K}$. Vector of prior weights for all topics $1, \ldots, K$ for document d
$\boldsymbol{\alpha}$	$\mathbb{R}_+^{K \times D}$	Matrix of $\alpha_{d,k}$ priors
β	$\mathbb{R}_+$	Uninformative prior weight. Same for all k and v
$\beta_{k,v}$	$\mathbb{R}_+$	Prior weight of term v in topic k
$w_{d,n}$	$1, \ldots, V$	Dictionary key / identifier of word n in document d
$z_{d,n}$	$1, \ldots, K$	Topic assignment for word n in document d
$\phi_{k,v}$	prob.: $[0, 1]$	Probability that word v occurs in topic k
$\boldsymbol{\phi}_k$	$\mathbb{R}^V$	Vector of multinomial probabilities s.t. $\sum_{v=1}^{V} \phi_{k,v} = 1$
$\boldsymbol{\phi}$	$\mathbb{R}^{K \times V}$	Matrix of $\phi_{k,v}$ probabilities
$\theta_{d,k}$	prob.: $[0, 1]$	Probability that document d belongs to topic k
$\boldsymbol{\theta}_d$	$\mathbb{R}^K$	Vector of multinomial probabilities s.t. $\sum_{k=1}^{K} \theta_{d,k} = 1$
$\boldsymbol{\theta}$	$\mathbb{R}^{D \times K}$	Matrix of $\theta_{d,k}$ probabilities
$n_{(d)}^k$	$\mathbb{N}_+$	Number of words in document d that carry label k, i.e. $n_{(d)}^k = \sum_{n=1}^{N_d} \mathbb{1}\{z_{n,d} = k\}$
$n_{(k)}$	$\mathbb{N}_+$	Number of words assigned to topic k. By definition $\sum_k n_{(k)} = N$
$n_{(k)}^v$	$\mathbb{N}_+$	Number of times word v is assigned to topic k, i.e. how often $\mathbb{1}\{z_{d,n} = k \;\&\; w_{d,n} = v\}$ equals one

Table 1: Definition of variables

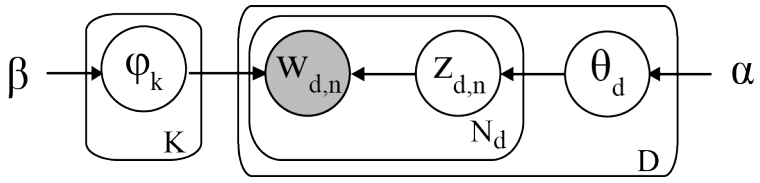

Figure 1: Graphical model: LDA

between the nodes in Figure 1 are as follows:

$$\boldsymbol{\phi}_k|\beta \sim \text{Dirichlet}_V(\beta) \quad (1)$$

$$\boldsymbol{\theta}_d|\alpha \sim \text{Dirichlet}_K(\alpha) \quad (2)$$

$$z_{d,n}|\boldsymbol{\theta}_d \sim \text{Categorical}_K(\boldsymbol{\theta}_d) \quad (3)$$

$$w_{d,n}|z_{d,n}, \boldsymbol{\phi} \sim \text{Categorical}_V(\boldsymbol{\phi}_{z_{d,n}}) \quad (4)$$

For an overview of all notational definitions, refer to Table 1.

As mentioned before, LDA is a generative probabilistic model. This means that the observed words are assumed to be a result of underlying latent distributions $\boldsymbol{\theta}$, $\boldsymbol{\phi}$ and $\mathbf{z}$. How exactly these distributions interact can be seen from the generative model:

1. For each topic $k = 1, 2, \ldots, K$:
   - Draw a distribution over words $\boldsymbol{\phi}_k|\beta \sim \text{Dir}_V(\beta)$
2. For each document $d = 1, 2, \ldots, D$:
   - Draw a topic mixture $\boldsymbol{\theta}_d|\alpha \sim \text{Dir}_K(\alpha)$
   - For each word $n = 1, 2, \ldots, N_d$ in document d:
     - Draw a word-topic assignment $z_{d,n}|\boldsymbol{\theta}_d \sim \text{Cat}_K(\boldsymbol{\theta}_d)$
     - Draw a word $w_{d,n}|z_{d,n}, \boldsymbol{\phi} \sim \text{Cat}_V(\boldsymbol{\phi}_{z_{d,n}})$

Note that the LDA setup thus far assumes scalar values for the hyperpriors α and β. In such a setting the hyperpriors are uninformative. Alternatively, α and β may be document- and/or topic-specific, respectively, making them informative. Extending to informative hyperpriors, however, will be postponed to Section 3.

The joint probability of all latent variables ($\boldsymbol{\theta}$, $\boldsymbol{\phi}$, $\mathbf{z}$) and observed variables ($\mathbf{w}$) in the LDA model is

$$p(\boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{z}, \mathbf{w}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \left(\prod_{k=1}^{K} p(\boldsymbol{\phi}_k|\boldsymbol{\beta})\right)\left(\prod_{d=1}^{D} p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})\left(\prod_{n=1}^{N_d} p(z_{d,n}|\boldsymbol{\theta}_d)p(w_{d,n}|z_{d,n}, \boldsymbol{\phi}_{z_{d,n}})\right)\right) \quad (5)$$

The interest of LDA, however, only indirectly involves the joint probability, because the main goal is to find the posterior of the latent distributions:

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

where the denominator requires integration over the full spaces of $\boldsymbol{\theta}$, $\boldsymbol{\phi}$ and $\mathbf{z}$. Even if an analytical solution existed, execution would be prohibitively expensive. Therefore, exact inference is not an option at this point and focus must be shifted towards *approximate inference*. Two main types of approximate inference are used in the context of LDA: Collapsed Gibbs sampling (Section 2.2) and variational inference (Section 2.3).

Key to the success of LDA is the choice for the conditional distributions: neighbouring nodes in the graphical model are always *conjugate pairs* (Section 2.1). In other words, the parent nodes that serve as a prior to their child nodes result in very convenient posterior distributions. This significantly eases the derivations of the conditional posterior distributions..

In Section 2.2 the collapsed Gibbs sampling approach by Griffiths and Steyvers (2004) for approximate inference will be introduced. Section 2.3 will start by introducing the concept of variational inference in general terms, including some necessary lemmata. Section 2.4 will then apply variational inference in the LDA setting.

2.1 Conjugate Priors

In the generative model, $\boldsymbol{\theta}_d$ serves as a prior for the distribution of $z_{d,n}|\boldsymbol{\theta}_d$. This section will demonstrate that using a Dirichlet distributed prior in a categorical distribution will lead to a Dirichlet distributed posterior. That is, the Dirichlet distribution is the conjugate prior of the categorical distribution. In order to do so, (2) and (3) will be formulated explicitly after which the distributions are combined to derive the distribution of the posterior and show conjugacy:

$$p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) = \frac{\Gamma\left(\sum_{j=1}^{K} \alpha_j\right)}{\prod_{j=1}^{K} \Gamma(\alpha_j)} \prod_{k=1}^{K} \theta_{d,k}^{\alpha_k - 1} \qquad \text{(Dirichlet prior)}$$

Extending the distribution of $z_{d,n}$ in (3) to $\mathbf{z}$ yields:

$$p(z_{d,n} = k|\boldsymbol{\theta}_d) = \prod_{k=1}^{K} \theta_{d,k}^{\mathbb{1}\{z_{d,n}=k\}}$$

$$p(\mathbf{z}|\boldsymbol{\theta}) = \prod_{d=1}^{D} \prod_{n=1}^{N_d} \prod_{k=1}^{K} \theta_{d,k}^{\mathbb{1}\{z_{d,n}=k\}} \qquad \text{(Categorical likelihood)}$$

Using the Dirichlet prior as the parameter in the categorical likelihood results in a Dirichlet posterior:

$$\begin{aligned}
p(\mathbf{z}|\boldsymbol{\theta}) \cdot p(\boldsymbol{\theta}|\boldsymbol{\alpha}) &= \prod_{d=1}^{D} \prod_{n=1}^{N_d} \prod_{k=1}^{K} \theta_{d,k}^{\mathbb{1}\{z_{d,n}=k\}} \underbrace{\frac{\Gamma\left(\sum_{j=1}^{K} \alpha_j\right)}{\prod_{j=1}^{K} \Gamma(\alpha_j)}}_{=B(\boldsymbol{\alpha})} \prod_{k=1}^{K} \theta_{d,k}^{\alpha_k - 1} \\
&= B(\boldsymbol{\alpha}) \prod_{d=1}^{D} \prod_{n=1}^{N_d} \prod_{k=1}^{K} \theta_{d,k}^{(\alpha_k - 1) + \mathbb{1}\{z_{d,n}=k\}} \\
&= B(\boldsymbol{\alpha}) \prod_{k=1}^{K} \theta_{d,k}^{(\alpha_k - 1) + \sum_{d=1}^{D} \sum_{n=1}^{N_d} \mathbb{1}\{z_{d,n}=k\}} \\
&\propto \mathrm{Dir}_K(\alpha_k + \sum_{d=1}^{D} \sum_{n=1}^{N_d} \mathbb{1}\{z_{d,n} = k\}) \\
&= \mathrm{Dir}_K(\alpha_k + n_{(k)}) \qquad \text{(Dirichlet posterior)}
\end{aligned}$$

The posterior stems from the same distributional family as its prior, so conjugacy has been shown. The final expression classically shows how the posterior distribution is determined by both the prior $\boldsymbol{\alpha}$ as well as the data $n_{(k)}$.

Analogously, the posterior of the prior-likelihood pair $p(\boldsymbol{\phi}_{z_{d,n}}|\boldsymbol{\beta})$ and $p(w_{d,n}|\boldsymbol{\phi}_{z_{d,n}})$ is

$$p(\boldsymbol{\phi}|\boldsymbol{\beta}) = \prod_{k=1}^{K} p(\boldsymbol{\phi}_k|\boldsymbol{\beta}) = \prod_{k=1}^{K} \frac{\Gamma(\sum_{v=1}^{V} \beta_v)}{\prod_{v=1}^{V} \Gamma(\beta_v)} \prod_{v=1}^{V} \phi_{k,v}^{\beta_v - 1}$$

$$p(\mathbf{w}|\boldsymbol{\phi}, \mathbf{z}) = \prod_{d=1}^{D} \prod_{n=1}^{N_d} \prod_{k=1}^{K} \prod_{v=1}^{V} \phi_{k,v}^{\mathbb{1}\{z_{d,n}=k \;\&\; w_{d,n}=v\}}$$

which results in the following Dirichlet posterior

$$\begin{aligned}
p(\mathbf{w}, \boldsymbol{\phi}|\beta) &\propto \mathrm{Dir}_V\left(\beta_v + \sum_{d=1}^{D} \sum_{n=1}^{N_d} \sum_{k=1}^{K} \mathbb{1}\{z_{d,n} = k \;\&\; w_{d,n} = v\}\right) \\
&= \mathrm{Dir}_V(\beta_v + n_{(k)}^{v})
\end{aligned}$$

2.2 Collapsed Gibbs Sampling

One way to deal with an unknown or complicated joint density like (5) is Markov chain Monte Carlo (henceforth MCMC), or more specifically (collapsed) Gibbs sampling (Geman and Geman, 1984). In MCMC, a Markov chain is constructed for which the equilibrium distribution has the properties of the target distribution. Repetitively resampling from this Markov chain is the essence of MCMC and will eventually result in a state from which samples are an asymptotically exact draw from the target distribution (Robert and Casella, 2005). Multiple rules have been developed by which the repetitive resampling is performed. One of these rules is called Gibbs sampling (Geman and Geman, 1984) in which the next state in the Markov chain is attained by sampling all variables, conditional on all other variables and the data. In other words, instances of every variable are sequentially drawn from their respective *full conditional distributions*.

Gibbs sampling is particularly useful when the target distribution is overly complicated, but its 'building blocks' (i.e. full conditionals) are known and simpler. This method was first introduced in the context of LDA by Griffiths and Steyvers (2004) and has played a big part in the development of LDA extensions, including the extensions presented in this thesis.

In order to get the full conditionals in the LDA framework, one needs to know the expected value of a Dirichlet distributed random variable:

Lemma 1 (Expected value of Dirichlet)**.** Let $\boldsymbol{\mu} \sim \text{Dir}_J(\boldsymbol{\alpha})$ then $\text{E}[\mu_k] = \frac{\alpha_k}{\sum_{j=1}^{J} \alpha_j}$.

As outlined in Griffiths and Steyvers (2004), only the topic-word assignments $z_{d,n}$ are sampled. Hence the full conditional of interest is

$$p(z_i = k | \mathbf{z}_{-i}, \mathbf{w}) \tag{6}$$

where the subscript $-i$ indicates that all values except the i-th value are considered. This probability distribution describes the probability that a given word w_i will be assigned to topic k, given that all data $\mathbf{w}$ and all other word-topic assignments $\mathbf{z}_{-i}$ are known. For notational convenience, the subscript referring to document d as well as explicitly conditioning on the hyperpriors α and β have been dropped. The difference in notation will be emphasised by introducing the subscript i, which will temporarily replace the subscript (d, n). By the end of the derivation, the usual notation will return and the result will be transformed to reflect document membership again.

Most terms that rely on available data $\mathbf{w}$ or terms that are part of the conditioning set will be removed over the course of the derivation. This allows for focus on relevant

dimensions of the distributions, while only affecting the characteristics of the distribution up to a multiplicative constant.

$$\begin{aligned} p(z_i = k|\mathbf{z}_{-i}, \mathbf{w}) &= \frac{p(z_i = k, w_i|\mathbf{z}_{-i}, \mathbf{w}_{-i}) \cdot p(\mathbf{z}_{-i}, \mathbf{w}_{-i})}{p(\mathbf{z}_{-i}, \mathbf{w})} \\ &\propto p(z_i = k, w_i|\mathbf{z}_{-i}, \mathbf{w}_{-i}) \\ &= p(w_i|z_i = k, \mathbf{w}_{-i}, \mathbf{z}_{-i}) \cdot p(z_i = k|\mathbf{w}_{-i}, \mathbf{z}_{-i}) \\ &= \underbrace{p(w_i|z_i = k, \mathbf{w}_{-i}, \mathbf{z}_{-i})}_{\text{=CGS: Term 1}} \cdot \underbrace{p(z_i = k|\mathbf{z}_{-i})}_{\text{=CGS: Term 2}} \qquad \text{(CGS: all terms)} \end{aligned}$$

The full conditional has been split in two distributions. The first may be interpreted as the probability that a word w_i is drawn, given that the word has been assigned to topic k. The second probability is broadly interpretable as the relative frequency of topic k in the entire corpus.

- **CGS: Term 1.** Further investigation of the first term leads to

$$p(w_i|z_i = k, \mathbf{w}_{-i}, \mathbf{z}_{-i}) = \int \underbrace{p(w_i|z_i = k, \boldsymbol{\phi}_k)}_{=\phi_{k,w_i}} \cdot \underbrace{p(\boldsymbol{\phi}_k|\mathbf{z}_{-i}, \mathbf{w}_{-i})}_{\text{see (8)}} d\boldsymbol{\phi}_k \tag{7}$$

  The second part of the integrand can be reformulated to a conjugate likelihood-prior structure for which the resulting posterior has been derived in Section 2.1

$$\begin{aligned} p(\boldsymbol{\phi}_k|\mathbf{z}_{-i}, \mathbf{w}_{-i}) &= \frac{p(\mathbf{w}_{-i}, \boldsymbol{\phi}_k|\mathbf{z}_{-i}) \cdot p(\mathbf{z}_{-i})}{p(\mathbf{z}_{-i}, \mathbf{w}_{-i})} \\ &\propto p(\mathbf{w}_{-i}|\boldsymbol{\phi}_k, \mathbf{z}_{-i}) \cdot p(\boldsymbol{\phi}_k) \\ &= \text{Dir}_V(n^v_{-i,(k)} + \beta) \end{aligned} \tag{8}$$

  The expression in (7) is therefore equal to the expected value of a Dirichlet distributed random variable. Using Lemma 1, the first term in (CGS: all terms) is equal to

$$\begin{aligned} p(w_i|z_i = k, \mathbf{w}_{-i}, \mathbf{z}_{-i}) &= \frac{n^{w_i}_{-i,(k)} + \beta}{\sum_{j \neq i} \left(n^{w_j}_{-i,(k)} + \beta \right)} \\ &= \frac{n^{w_i}_{-i,(k)} + \beta}{n_{-i,(k)} + V\beta} \end{aligned} \qquad \text{(CGS: Term 1)}$$

- **CGS: Term 2.** Analogous reasoning will be applied to the second term:

$$p(z_i = k|\mathbf{z}_{-i}) = \int \underbrace{p(z_i = k|\boldsymbol{\theta}_d)}_{\theta_{d,k}} \cdot \underbrace{p(\boldsymbol{\theta}_d|\mathbf{z}_{-i})}_{\text{see (10)}} d\boldsymbol{\theta}_d \tag{9}$$

The second part of the integrand can be reformulated as a conjugate likelihood-prior pair for which the posterior is known from Section 2.1

$$\begin{aligned} p(\boldsymbol{\theta}_d|\mathbf{z}_{-i}) &= \frac{p(\mathbf{z}_{-i}|\boldsymbol{\theta}_d) \cdot p(\boldsymbol{\theta}_d)}{p(\mathbf{z}_{-i})} \\ &\propto p(\mathbf{z}_{-i}|\boldsymbol{\theta}_d) \cdot p(\boldsymbol{\theta}_d) \\ &= \mathrm{Dir}_K(n^k_{-i,(d)} + \alpha) \end{aligned} \tag{10}$$

Hence (9) is the expectation of a Dirichlet distributed random variable, which is the final expression for the second term in (CGS: all terms)

$$p(z_i = k|\mathbf{z}_{-i}) = \frac{n^k_{-i,(d)} + \alpha}{n_{-i,(d)} + K\alpha} \tag{CGS: Term 2}$$

Inserting (CGS: Term 1) and (CGS: Term 2) into (CGS: all terms) yields the full conditional that was first formulated in Griffiths and Steyvers (2004):

$$p(z_i = k|\mathbf{z}_{-i}, \mathbf{w}) = \frac{n^{w_i}_{-i,(k)} + \beta}{n_{-i,(k)} + V\beta} \cdot \frac{n^k_{-i,(d)} + \alpha}{n_{-i,(d)} + K\alpha} \tag{11}$$

Intuitively, this may be interpreted as the product of two empirical probabilities: The probability that word w_i occurs in topic k multiplied by the probability hat topic k occurs in document d. This expression will play a major role in the dynamics of the LDA extensions in this paper.

At this point, the rule by which samples are repetitively drawn in the Monte Carlo procedure has been established: every draw is performed on the full conditional. The full conditional of every z_i is dependent on $\boldsymbol{z}_{-i}$ so the rule must constantly be adapted to the current state of the Markov chain. Once all z_i are assigned a value between 1 and K, the Markov chain is initialised, the iterative sampling procedure according to the ever-changing full conditional distribution of the z_i's can be started.

A sample from the posterior distribution $p(\mathbf{z}|\mathbf{w})$ is generated in every iteration of the Markov chain. Latent distributions $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ can be estimated from every one of these iterations:

$$\hat{\phi}_{k,v} = \frac{n^v_{(k)} + \beta}{n_{(k)} + V\beta}$$

$$\hat{\theta}_{d,k} = \frac{n^k_{(d)} + \alpha}{n_{(d)} + K\alpha}$$

After a burn-in period of iterations, every t-th sample (Markov state) can be saved. Taking averages for each $\hat{\boldsymbol{\phi}}$ and $\hat{\boldsymbol{\theta}}$ in the saved Markov states, results in the "predictive distributions

over new words w and topics z conditioned on $\mathbf{w}$ and $\mathbf{z}$" (Griffiths and Steyvers, 2004, p. 5230).

So far the *collapsed* aspect of this Gibbs sampling procedure has not been addressed explicitly. The document-topic distribution $\boldsymbol{\theta}$ and topic-word distributions $\boldsymbol{\phi}$ serve as (Dirichlet) priors to the categorical likelihood $\mathbf{z}$. As can be seen in equations (7) and (9), these priors have been integrated (collapsed) out. The collapsing out results in an unconditional distribution (independent of its prior) from which samples will be taken. In case of conjugate pairs, the resulting distribution has a particularly simple form in comparison to the initial conditional distributions, thus making the Gibbs sampling process less complex. Also, dependencies between all categorical variables that relied on the Dirichlet prior are created. This can be easily confirmed by observing that the distribution of z_i is partially determined by $\mathbf{z}_{-i}$ via the n-terms in (11).

2.3 Variational Inference: Background

The posterior distribution of all latent distributions $p(\boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{z}|\mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ is to be computed, but is intractable. Variational inference can be used to deal with models for which it is infeasible to evaluate the posterior distribution. Infeasibility may stem from the high dimensionality of latent spaces or complex forms that disable analytical tractability (Bishop, 2006, p. 461). By using the tools of variational inference, the posterior distribution can be approximated by more handleable distributions that in turn can be optimised. The general idea is to minimise the Kullback-Leibler divergence (Kullback and Leibler, 1951) between the true posterior and the approximate distribution, whilst restricting the family of distributions from which we can select the approximate distribution. The family must be rich enough to resemble relevant features of the true distribution, while simultaneously be restricted to family members that are tractable and feasible for optimisation.

Assume a model with observed variables $\mathbf{x}$ and latent variables $\mathbf{y}$ such that the posterior distribution $p(\mathbf{y}|\mathbf{x})$ is intractable, due the high dimensionality of the latent space in which $\mathbf{x}$ resides. The aim of variational inference is to approximate $p(\mathbf{y}|\mathbf{x})$ with a tractable density $q(\mathbf{y})$. To do so, the Kullback-Leibler divergence is minimised

$$q^*(\mathbf{y}) = \arg\min_{q(\mathbf{y})\in\mathcal{D}} \mathrm{KL}\left(q(\mathbf{y})||p(\mathbf{y}|\mathbf{x})\right) \tag{12}$$

where

$$\begin{aligned}\mathrm{KL}\left(q(\boldsymbol{y})||p(\boldsymbol{y}|\boldsymbol{x})\right) &= \mathrm{E}_q\left[\log\frac{q(\mathbf{y})}{p(\mathbf{y}|\mathbf{x})}\right] = \mathrm{E}_q\left[\log q(\mathbf{y})\right] - \mathrm{E}_q\left[\log p(\mathbf{y}|\mathbf{x})\right] \\ &= \mathrm{E}_q\left[\log q(\mathbf{y})\right] - \mathrm{E}_q\left[\log p(\mathbf{y},\mathbf{x})\right] + \cancel{\log p(\mathbf{x})}\end{aligned}$$

and where $\mathcal{D}$ is the family of distributions from which we can select $q(\mathbf{y})$. The above still involves $p(\boldsymbol{x})$, however it is nothing more than an additive constant in the optimisation function (12), so it may be ignored.

Since the KL-divergence is non-negative (Kullback and Leibler, 1951), it is easy to verify that $\log p(\boldsymbol{x})$ is lower bounded by $\mathrm{E}_q\left[\log p(\mathbf{y},\mathbf{x})\right] - \mathrm{E}_q\left[\log q(\mathbf{y})\right]$. Because $\log p(\boldsymbol{x})$ is also referred to as the *evidence* in Bayesian statistics, the following is called Evidence Lower Bound (ELBO):

$$\mathrm{ELBO}(q) = \mathrm{E}_q\left[\log p(\mathbf{y},\mathbf{x})\right] - \mathrm{E}_q\left[\log q(\mathbf{y})\right] \tag{13}$$

From the definition of the KL-divergence and ELBO(q), it can be seen that minimising the KL-divergence w.r.t. $q(\mathbf{y})$ is equivalent to maximising ELBO(q) which is what the focus will be on henceforth.

If no restrictions are placed on the family of distributions $\mathcal{D}$ from which $q(\mathbf{y})$ can be selected, the optimal choice would be to set $q(\mathbf{y})$ equal to $p(\mathbf{y}|\mathbf{x})$, because the KL-divergence would obviously be zero. To ensure simplicity in the structure of the variational distribution, $q(\mathbf{y})$ is limited to distributions that factorise over all its marginal distributions. In other words, the joint variational distribution of $\mathbf{y} = (y_1, y_2, \ldots, y_M)$ is equal to the product of the marginal variational distributions

$$q(\mathbf{y}) = \prod_{m=1}^{M} q_m(y_m)$$

This is known as the mean-field variational family, where each latent variable is represented by its own variational factor $q_m(y_m)$, that may be considered as factorised marginal distributions (Blei et al., 2017; Jordan et al., 1999).

In general, a joint distribution $p(\mathbf{x},\mathbf{y})$ (such as the one in (13)) can be factorised like

$$\log p(\mathbf{y},\mathbf{x}) = \log\left[p(\mathbf{x})\prod_{m=1}^{M} p(y_m|\mathbf{y}_{-m},\mathbf{x})\right] = \log p(\mathbf{x}) + \sum_{m=1}^{M}\log p(y_m|\mathbf{y}_{-m},\mathbf{x})$$

Incorporating the mean-field variational family assumption in ELBO(q) now yields

$$\mathrm{ELBO}(q) = \log p(\mathbf{x}) + \sum_{m=1}^{M}\left(\mathrm{E}_q\left[\log p(y_m|\mathbf{y}_{-m},\mathbf{x})\right] - \mathrm{E}_{q_m}(\log q_m(y_m))\right)$$

Recall that the aim is to maximise ELBO(q) with respect to the variational distributions, hence $\log p(\mathbf{x})$ may be regarded as a constant. Since the ELBO(q) will be maximised and eventually the derivative w.r.t. $q_m(y_m)$ will be set to zero, a convenient formulation makes all dependencies on $q_m(y_m)$ explicit. Using the law of iterated expectations, taking the partial derivative and solving for $q_m(y_m)$ yields

$$\begin{aligned}
\text{ELBO}(q)_m &= \int q_m(y_m)\, \mathrm{E}_{-m}\left[\log p(y_m|\mathbf{y}_{-m}, \mathbf{x})\right] dy_m - \int q_m(y_m) \log q_m(y_m) dy_m \\
\frac{\partial \text{ELBO}(q)_m}{\partial q_m(y_m)} &= \mathrm{E}_{-m}\left[\log p(y_m|\mathbf{y}_{-m}, \mathbf{x})\right] - \log q_m(y_m) - 1 \stackrel{!}{=} 0 \\
q_m^*(y_m) &\propto \exp\left[\mathrm{E}_{-m}(\log p(y_m|\mathbf{y}_{-m}, \mathbf{x}))\right] \qquad (14)
\end{aligned}$$

This expression can be solved when $p(y_m|\mathbf{y}_{-m}, \mathbf{x})$ has been derived, which will be done in Section 2.4.

Before moving on to the general solutions for exponential family members, it is interesting to pay close attention to the similarities between Gibbs sampling and (coordinate ascent) variational inference. They are more closely related than one may suspect at first sight. Recall that Gibbs sampling uses the full conditional distribution to sample from. The coordinate ascent variational inference method discussed here uses the (exponentiated) expected (log) value of that full conditional (see (14)) to set each variational factor (Blei et al., 2017).

In addition to (14), there is another useful relation between the optimal variational factors $q_m^*(y_m)$ and the full conditionals. If the full conditional is a member of the exponential family, then $q_m^*(y_m)$ is member of that same family, but with different *natural parameters* $\boldsymbol{\nu}_m^*$ (see Appendix A for a proof). The only difference (up to a multiplicative constant) between the true full conditionals and the variational factors $q_m(y_m)$ lies in the natural parameters, which is the reason why in this setting they are also referred to as the *variational parameters* of $q_m(y_m)$:

$$\boldsymbol{\nu}_m^* \propto \mathrm{E}_{q_{-m}}\left[\eta_m(\mathbf{x}, \mathbf{y}_{-m})\right] \qquad (15)$$

where $\eta_m(\mathbf{x}, \mathbf{y}_{-m})$ is the natural parameter of the complete conditional. For a full derivation, refer to Appendix A.

In order to arrive at the variational factors in the LDA model, it is useful to be aware of the following lemmata:

Lemma 2 (Markov blankets)**.** LDA is a statistical model that can be expressed as a graphical model (see Figure 1). Such Bayesian networks posses a number of properties that help in identifying conditional independence of nodes in the network. Awareness of these properties

allows one to nimbly assess whether the conditioning set of variables can be reduced or not, without affecting the conditional distribution. The Markov blanket of a node A - denoted as $MB(A)$ - in a Bayesian network consists of

1. all parents of A
2. all children of A and
3. all parents of the children of A (so called co-parents of A).

All nodes outside of $MB(A)$ are conditionally independent of A if conditioned on $MB(A)$. This implies that all nodes outside of A's Markov Blanket are redundant in the conditioning set as long as $MB(A)$ is contained in the conditioning set (Bishop, 2006).

Lemma 3 (D-separation due to collider node)**.** A path between nodes A and C in a Bayesian network is d-separated (i.e. blocked) by a node set Z if both nodes A and C are directed towards B, and neither B nor any of its descendants are part of the set Z (Geiger et al., 1990).

Lemma 4 (Expected value of log-Dirichlet)**.** Let $\mathbf{x} \sim \mathrm{Dir}_J(\boldsymbol{\alpha})$, then the expected value of the logarithm of x_k equals

$$\mathrm{E}(\log x_k) = \psi(\alpha_k) - \psi\left(\sum_{j=1}^{J} \alpha_j\right)$$

where $\psi(y) = \frac{d}{dy} \log \Gamma(y)$ is the digamma function.

Lemma 5 (Natural parameters - Dirichlet distribution)**.** Let $\mathbf{x} \in \mathbb{R}^M$ and $\mathbf{x} \sim \mathrm{Dir}(\boldsymbol{\alpha})$, then the natural parameter of $\mathbf{x}$ is

$$\eta(\mathbf{x}, \boldsymbol{\alpha}) = (\alpha_1, \ldots, \alpha_M)$$

Lemma 6 (Natural parameters - categorical distribution)**.** Let $x \in \{1, \ldots, N\}$ and $x \sim \mathrm{Cat}_N(\mathbf{p})$, then the natural parameter of x is

$$\eta(x, \mathbf{p}) = (\log p_1, \log p_2, \ldots, \log p_N)$$

2.4 Variational Inference in Latent Dirichlet Allocation

In order to use (15) and obtain the optimal variational parameters, the full conditionals of the latent distributions are required. Three groups of latent distributions exist in LDA: $\mathbf{z}$, $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$. Each of them has a different full conditional distribution. This section will start off

with obtaining the true full conditionals (Section 2.4.1) and will afterwards proceed with the identification of the optimal variational distributions (Section 2.4.2). As a final result, we will end up with the Coordinate Ascent Variational Inference (CAVI) algorithm for LDA.

2.4.1 Full Conditionals in LDA

Getting to the full conditionals will build on the conjugacy results of Section 2.1 and on the lemmata presented at the end of Section 2.3.

1. **Full conditional of document-topic distributions:** The Markov blanket $MB(\boldsymbol{\theta}_d)$ consists of the parents α and the children $\mathbf{z}_d$ (there are no co-parents for $\boldsymbol{\theta}_d$), so

$$p(\boldsymbol{\theta}_d|\mathbf{z}, \boldsymbol{\theta}_{-d}, \boldsymbol{\phi}, \mathbf{w}) = p(\boldsymbol{\theta}_d|\mathbf{z}_d, \alpha)$$

   where it is known that $\mathbf{z}_d \sim \prod_{n=1}^{N_d} \text{Cat}(\boldsymbol{\theta}_d)$ and $\boldsymbol{\theta}_d|\alpha \sim \text{Dir}(\alpha)$ from the LDA model specification. By the conjugacy results from Section 2.1, the full conditional of topic proportions $p(\boldsymbol{\theta}_d|\mathbf{z}_d, \alpha) \sim \text{Dir}(\boldsymbol{\alpha}_d^*)$, where

$$\begin{aligned} \boldsymbol{\alpha}_d^* &= \left(\alpha + \sum_{n=1}^{N_d} \mathbb{1}\{z_{d,n} = 1\}, \ldots, \alpha + \sum_{n=1}^{N_d} \mathbb{1}\{z_{d,n} = K\} \right) \\ &= \left(\alpha + n_{(d)}^1, \ldots, \alpha + n_{(d)}^K \right) \end{aligned} \tag{16}$$

2. **Full conditional of topic-words distributions:** The Markov blanket $MB(\boldsymbol{\phi}_k)$ consists of the children $\mathbf{z}$, the co-parents $\boldsymbol{\phi}_{-k}$ and $\mathbf{w}$, and parents β, so

$$\begin{aligned} p(\boldsymbol{\phi}_k|\mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\phi}_{-k}, \mathbf{w}, \beta) &= p(\boldsymbol{\phi}_k|\mathbf{z}, \boldsymbol{\phi}_{-k}, \mathbf{w}, \beta) \\ &\propto p(\boldsymbol{\phi}_k, \mathbf{w}|\mathbf{z}, \boldsymbol{\phi}_{-k}, \beta) \\ &= p(\mathbf{w}|\boldsymbol{\phi}_k, \boldsymbol{\phi}_{-k}, \mathbf{z}, \beta) \cdot p(\boldsymbol{\phi}_k|\boldsymbol{\phi}_{-k}, \mathbf{z}, \beta) \\ &= p(\mathbf{w}|\boldsymbol{\phi}, \mathbf{z}) \cdot p(\boldsymbol{\phi}_k|\beta) \end{aligned}$$

   where the last term in the final equation follows from Lemma 3: $w_{d,n}$ is the colliding node between $\boldsymbol{\phi}_k$ and $\mathbf{z}$, but $w_{d,n}$ is not part of the conditioning set and therefore they are conditionally independent. From the LDA model specification it is known that $p(\mathbf{w}|\boldsymbol{\phi}, \mathbf{z}) \sim \prod_{d=1}^{D} \prod_{n=1}^{N_d} \text{Cat}(\boldsymbol{\phi}_{z_{d,n}})$ and $p(\boldsymbol{\phi}_k|\beta) \sim \text{Dir}(\beta)$. Considering that the latter serves as a conjugate prior to the former distribution, it is known (see Section 2.1) that the result must follow a $\text{Dir}(\boldsymbol{\beta}_k^*)$ distribution, where

$$\boldsymbol{\beta}_k^* = \left(\beta + \sum_{d=1}^{D} \sum_{n=1}^{N_d} \mathbb{1}_{\{z_{d,n}=k \ \& \ w_{d,n}=1\}} \quad , \ldots, \quad \beta + \sum_{d=1}^{D} \sum_{n=1}^{N_d} \mathbb{1}_{\{z_{d,n}=k \ \& \ w_{d,n}=V\}} \right) \tag{17}$$

3. **Full conditional of word assignments:** The Markov blanket $MB(z_{d,n})$ consists of the parents $\boldsymbol{\theta}_d$, the children $w_{d,n}$ and the co-parents $\boldsymbol{\phi}_k$. Hence

$$p(z_{d,n} = k|\mathbf{z}_{-d,n}, \boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{w}) = p(z_{d,n} = k|\boldsymbol{\theta}_d, \boldsymbol{\phi}_k, w_{d,n})$$

This is a conditional distribution that is not known directly, but using the chain rule the above equation can be written like

$$\begin{aligned} p(z_{d,n} = k|\boldsymbol{\theta}_d, \boldsymbol{\phi}, w_{d,n}) &\propto p(z_{d,n} = k, w_{d,n} = v|\boldsymbol{\theta}_d, \boldsymbol{\phi}) \\ &= p(w_{d,n} = v|z_{d,n}, \boldsymbol{\theta}_d, \boldsymbol{\phi}_{z_{d,n}}) \cdot p(z_{d,n} = k|\boldsymbol{\theta}_d, \boldsymbol{\phi}) \\ &= p(w_{d,n} = v|z_{d,n}, \boldsymbol{\phi}) \cdot p(z_{d,n} = k|\boldsymbol{\theta}_d) \\ &= \phi_{k,v} \cdot \theta_{d,k} \end{aligned} \tag{18}$$

where $\boldsymbol{\theta}_d$ is cancelled because it is not part of the $MB(w_{d,n})$ and $\boldsymbol{\phi}$ is cancelled because it is d-separated from $z_{d,n}$ by the node $w_{d,n}$ (Lemma 3)

2.4.2 Variational Factors in LDA

With the results from the previous section, we now know the distributional family of each variational factor, but the parameters for those distributions are still unknown. Let the unknown parameters be $\boldsymbol{\gamma}_d$, $\boldsymbol{\lambda}_k$ and $\boldsymbol{\psi}_{d,n}$ for the document-topic, topic-word and word-assignment distributions, respectively:

1. Variational distribution of document-topic distribution: $q_{\boldsymbol{\theta}_d}(\boldsymbol{\theta}_d) = \text{Dir}(\boldsymbol{\gamma}_d)$. The natural parameters, $\eta_{\theta_d}(\alpha, \boldsymbol{z}_d)$, of the full conditional are $(\alpha + n^k_{(d)})$ for $k = 1, \ldots, K$.

2. Variational distribution of topic-word distribution: $q_{\boldsymbol{\phi}_k}(\boldsymbol{\phi}_k) = \text{Dir}(\boldsymbol{\lambda}_k)$. The natural parameters, $\eta_{\phi_k}(\beta, \boldsymbol{z}_{d,n})$, of the full conditional are $\beta + \sum_{d=1}^{D} \sum_{n=1}^{N} \mathbb{1}_{\{z_{d,n}=k \ \& \ w_{d,n}=v\}}$ for $v = 1, \ldots, V$.

3. Variational distribution of word assignment distribution: $q_{\boldsymbol{z}_{d,n}}(\boldsymbol{z}_{d,n}) = \text{Cat}(\boldsymbol{\psi}_{d,n})$. By Lemma 6, the natural parameters, $\eta_{z_{d,n}}(\phi_{z_{d,n}}, \boldsymbol{\theta}_d, w_{d,n})$, of the full conditional are equal to the logarithm of the parameters. Therefore $\eta_{z_{d,n}}(\phi_{z_{d,n}}, \boldsymbol{\theta}_d, w_{d,n}) = \log \phi_{k,v} \log \theta_{d,k}$ for $k = 1, 2, \ldots, K$.

The natural parameters of the full conditionals will be crucial in this section, since they have been linked explicitly with the optimal natural parameters of the variational factors (see expression (15)).

1. **Variational distribution of document-topic distributions:** To find the appropriate parameters $\boldsymbol{\gamma}_d$ for the variational distribution, we will rely on (15):

$$\boldsymbol{\gamma}_d^* \propto \mathrm{E}_{q_{-\boldsymbol{\gamma}_d}}[\eta_{\theta_d}(\alpha, \boldsymbol{z}_d)]$$

The reason α and $\boldsymbol{z}_d$ appear in the expression is that they constitute the Markov blanket that was used in the full conditional.

The expectation is taken w.r.t. all variational distributions, except $\boldsymbol{\gamma}_d$. Since the term inside the expectation operator, i.e. the natural parameters of the full conditional, contains only α and $\boldsymbol{z}_d$, the expected value is equivalent to $\mathrm{E}_{z_d}[\eta_{\theta_d}(\alpha, \boldsymbol{z}_d)]$. We have

$$\begin{aligned}
\gamma_{d,k}^* &\propto \mathrm{E}_{q_{z_{d,k}}}[\eta_{\theta_{d,k}}(\alpha, \boldsymbol{z}_d)] \\
&= \alpha + \sum_{n=1}^{N_d} \mathrm{E}_{q_{z_d}}(\mathbb{1}\{z_{d,n} = k\}) \\
&= \alpha + \sum_{n=1}^{N_d} \psi_{d,n,k}
\end{aligned}$$

where the final line follows from the fact that q_{z_d} follows a categorical distribution with the (unknown) parameters $\psi_{d,n,k}$ for $n = 1, \ldots, N_d$ and $k = 1, \ldots, K$. The document-topic variational distribution is $q_{\theta_{d,k}}^* = \mathrm{Dir}(\gamma_{d,k}^*)$.

2. **Variational distribution of topic-words distributions:** Analogous to the derivation for the variational distribution of the document-topic distributions, we know that

$$\begin{aligned}
\lambda_{k,v}^* &\propto \mathrm{E}_{q_{-\boldsymbol{\lambda}_k}}(\eta_{\phi_k}(\beta, z_{d,n})) \\
&= \mathrm{E}_{q_{z_{d,n}}}(\eta_{\phi_k}(\beta, \boldsymbol{z})) \\
&= \left(\beta + \sum_{d=1}^{D}\sum_{n=1}^{N_d} \mathrm{E}_{q_{z_{d,n}}}[\mathbb{1}_{z_{d,n}=k}]\mathbb{1}_{w_{d,n}=v}\right) \\
&= \left(\beta + \sum_{d=1}^{D}\sum_{n=1}^{N_d} \psi_{d,n,k} \cdot \mathbb{1}_{w_{d,n}=v}\right)
\end{aligned}$$

Therefore, the variational distribution of the topic-words distributions is $q_{\boldsymbol{\phi}_{k,v}}^* = \mathrm{Dir}(\lambda_{k,v}^*)$.

3. **Variational distribution of word assignments:** By expression (16), the natural parameters of the variational distribution of $z_{d,n}$ are

$$\boldsymbol{\nu}_{q_{z_{d,n}}}^* \propto \mathrm{E}_{-\boldsymbol{\psi}_{d,n}}[\eta_{z_{d,n}}(\phi_{z_{d,n}}, \boldsymbol{\theta}_d, w_{d,n})]$$

Since $q_{z_{d,n}}(\boldsymbol{\psi}_{d,n})$ is a categorical distribution, the natural parameters are the logarithm of the parameters (see Lemma 6). The actual parameters of $q_{z_{d,n}}(\boldsymbol{\psi}_{d,n})$ are denoted as

$\boldsymbol{\psi}_{d,n}$:

$$\begin{aligned}
\boldsymbol{\psi}^*_{d,n,k} = \exp\left(\boldsymbol{\nu}^*_{q_{z_{d,n}}}\right) &\propto \exp\left(\mathrm{E}_{-\boldsymbol{\psi}_{d,n}}[\eta_{z_{d,n}}(\phi_{z_{d,n}}, \boldsymbol{\theta}_d, w_{d,n})]\right) \\
&= \exp\left(\mathrm{E}_{-\boldsymbol{\psi}_{d,n}}[\log \phi_{k,v} + \log \theta_{d,k}]\right) \\
&= \exp\left(\mathrm{E}_{q_{\phi_k}}[\log \phi_{k,v}] + \mathrm{E}_{q_{\theta_d}}[\log \theta_{d,k}]\right) \\
&= \exp\left(\Psi(\lambda_{k,v}) - \Psi(\sum_{u=1}^{V} \lambda_{k,u}) + \Psi(\gamma_{d,k}) - \Psi(\sum_{j=1}^{K} \gamma_{d,j})\right)
\end{aligned}$$

where the last line follows from Lemma 4.

3 Incorporating Available Information

The input required to perform baseline LDA is merely a corpus with documents, the number of topics K and a value for the hyperpriors α and β. The corpus used in this paper has an additional source of information that can be exploited, namely the JEL codes for each document. Not only are the documents labeled, the structure of the labels is also known. For example, the labels D12 and D01 are both members of the label family D and are likely to have more in common than labels A14 and Q21. By incorporating such prior knowledge, the unsupervised baseline LDA can be transformed into a (semi)supervised method.

This section will present three extensions to the baseline LDA model: Labeled LDA (L-LDA), Hierarchically Supervised LDA (HSLDA) and CascadeLDA. Each of the extensions utilises the available information in its own way and the quality of their results depends on the ultimate aim of the user and at times on very specific features of the dataset.

3.1 Extension 1: Labeled LDA

The aim of this paper is to build a classifier that is able to recommend JEL codes for previously unseen academic abstracts and full academic papers. This implies that the interest is not to identify latent topics, but rather explicit topics that have a one-to-one correspondence with the JEL codes. There are multiple ways to manipulate the topics identified by LDA. One could, for example, provide a set of seed words and use them to bias topic-word distributions towards certain terms (Jagarlamudi et al., 2012). Alternatively, by creating Must-Links and Cannot-Links between words and topics (Andrzejewski et al., 2009) the structure of topic-word distributions is restricted in a favourable way. These methods can be used if the dataset is an unlabeled corpus, but the researchers have some domain knowledge about the topics they wish to identify. External knowledge sources such as Wikipedia articles regarding the

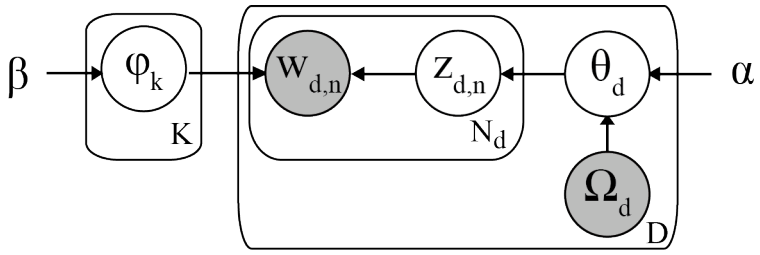

Figure 2: Graphical model: Labeled LDA

desired topics can also be used as a form of prior knowledge (Wood et al., 2016). In the case of this research, we can manipulate the topics to exactly correspond with the labels. Such a setting of explicit topics in labeled data has been investigated in Ramage et al. (2009) and Rubin et al. (2012). A crucial piece of information is that the topics/labels covered in a every document is known a priori and more importantly, it is known which topics are *not* part of the document. To put the above in more LDA-specific terms: It is known which of the K elements in $\boldsymbol{\theta}_d$ have a positive document-topic mixture and which elements are equal to zero. Hence, the space in which the topic-document distribution resides (during training) for every document can be shrunk tremendously, since it is limited to only the JEL-codes attached to that document. This causes a shift in the way the conditional distributions in the model interact through a three step process: Firstly, the parameters for the distribution of $z_{d,n}$ (recall that $z_{d,n} \sim \text{Cat}_K(\boldsymbol{\theta}_d)$) are mostly equal to zero, with positive values only in the positions that correspond to the labels of the document. This leads to the second step of the process: $\mathbf{z}_{d,1:N}$ can only be assigned to the positively loaded labels, hence all words $\mathbf{w}_{d,1:N}$ in that document will be associated with those labels. As a result, document d only affects those topic-word distributions $\boldsymbol{\phi}_k$ that correspond to the topics (labels) of that document.

The above has been dubbed Labeled LDA or L-LDA by Ramage et al. (2009), who claims that L-LDA outperforms support vector machines in identifying label-specific document snippets and is competitive when it comes to discriminatively assigning labels to unseen documents. Despite some contextual differences, L-LDA is in practice equivalent to Rubin's Flat-LDA (Rubin et al., 2012).

Figure 2 illustrates how the label information is incorporated to extend the baseline LDA model. The full generative model of L-LDA is nearly identical to LDA, except for the document-topic distribution $\boldsymbol{\theta}_d$:

$$\boldsymbol{\theta}_d \quad \sim \quad \text{Dir}_K(\alpha \cdot \boldsymbol{\Omega}_d)$$

where $\boldsymbol{\Omega}_d$ is a K-dimensional 0-1 vector that indicates whether label k is part of document

Table 2: The ten most likely words for five JEL codes

ROOT	E32	D82	J24	R31	I12
econom	cycl	agent	skill	hous	mortal
cours	model	inform	capit	price	health
signific	busi	princip	human	boom	increase
student	shock	optim	educ	transact	percent
research	recess	mechan	labor	home	age
author	aggreg	privat	school	area	birth
regress	volatil	effici	worker	buyer	suggest
expenditur	rate	incent	occup	percent	rate
call	fluctuat	alloc	wage	seller	life
yield	product	select	college	rent	effect

ROOT is the generic label that is assigned to all documents to improve classification results of real labels

E32: Macroeconomics & Monetary Economics → Prices, Business Fluctuations & Cycles → Business Fluctuations - Cycles

D82: Microeconomics → Information, Knowledge & Uncertainty → Asymmetric & Private Information - Mechanism Design

J24: Labour & Demographic Economics → Demand & Supply of Labour → Human Capital - Skills - Occupational Choice - Labour Productivity

R31: Urban, Rural, Regional, Real Estate & Transportation Economics → Real Estate Markets, Spatial Production Analysis & Firm Location →Housing Supply and Markets

I12: Health, Education & Welfare → Health → Health Behaviour

d's label set. This is the dynamic that forces all topic-loadings that are *not* part of the document's label set to zero while keeping the rest positive. It results in highly accurate topic-word distributions for each label as can be seen in Table 2.

An important characteristic is the ROOT label which is assigned to every single document. The purpose of this label is to capture generic (academic) language and thus clean these words from the actual JEL labels.

Shrinking a document's topic-document distribution sets the stage for a noteworthy side effect. Due to the fewer possible values, the iterative processes of Gibbs sampling and CAVI during training time converge much faster and in a more stable way.

3.2 Extension 2: Hierarchically Supervised LDA

The labels of the documents have an implicit structure which is illustrated in Figure 3. The labels consist of one letter and two digits, like D42 and F31, where the letter stands for a general category (level 1) such as microeconomics or international finance. The first digit represents the first refinement within the general category (level 2) and the second digit provides the final level of granularity (level 3 or leaf label). Therefore, it is to be expected that JEL code D42[1] has a topic-word distribution very closely related to the JEL code D41[2]. More generally, neighbouring leaf labels tend to be highly similar and are, also for the human reader, hard to distinguish between.

The idea of related topics within the LDA model has been investigated in a number of papers (e.g. Blei and Lafferty (2006)), but these generally do not allow for pre-specified correlations between the topics, whereas our dataset would require so. Additionally, the Semi-Supervised Hierarchical Topic Model (SSHLDA) by Mao et al. (2012) introduces a promising model which learns new topics automatically. Since the aim of this thesis is to classify according to pre-specified label and not to identify new topics, SSHLDA would unnecessarily complicate computations and derivations in this thesis, without any guarantee regarding the correctness of the hierarchy. Dependency-LDA introduced by Rubin et al. (2012) does account for dependencies between labels while also allowing for multiple labels per document. However, Dependency-LDA would require creating positive loadings to all of the topics that are related to the document's labels which in turn decreases the discriminative power of the model. Additionally, it is specifically designed to perform well on corpora with a large number of labels whose frequencies follow a power law distribution. Since Dependency-LDA outperforms other methods only for datasets with labels that follow a power law distribution, this method has also not been pursued further.

Another promising model is found in Hierarchically Supervised LDA (HSLDA) by Perotte et al. (2011) for which performing out-of-sample predictions is the primary aim. The baseline LDA model is augmented with an entirely new mechanism that exploits the hierarchy structure of the labels via $z_{d,n}$. Figure 4 shows the graphical model for HSLDA. The top part of the model is a baseline LDA model in which actual *latent* topics are searched for. The information on the document labels is integrated in the bottom part of the graphical model.

[1] D42: Microeconomics → Market structure, pricing → Monopoly

[2] D41: Microeconomics → Market structure, pricing → Perfect competition

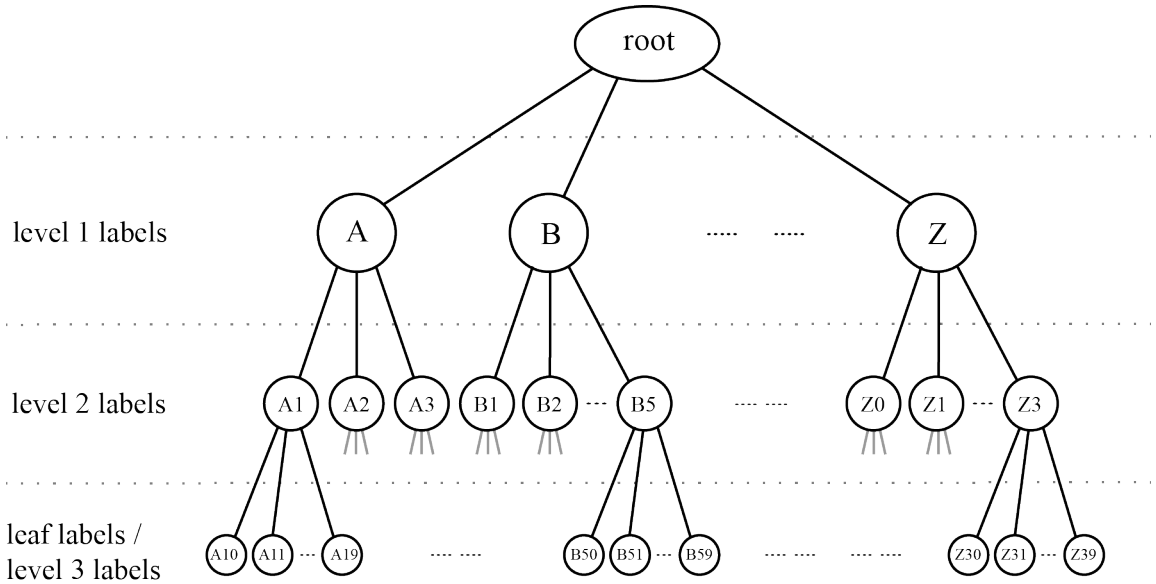

Figure 3: Label structure: JEL code tree

3.2.1 Generative Model

Before diving into the technical parts, a more intuitive explanation of the model's mechanics will be provided here. The most intuitive take on this model is to start off at node $a_{root,d}$. This node serves as a running variable in a probit model and follows a normal distribution with mean $\bar{z}_1\eta_{root,1}+\bar{z}_2\eta_{root,2}+\cdots+\bar{z}_K\eta_{root,K}$, where $\bar{\mathbf{z}}_k$ represents the current share of topic k in the document-topic mixture. During testing a given label will be assigned to a document only if its corresponding running variable exceeds a pre-specified threshold. If that is the case, the label dummy $y_{l,d}$ for label l is set to one. The running variables further down the label tree are dependent on the other label dummies. During testing they are truncated from above at the threshold if its parent has a value below the threshold. It is therefore impossible for a label down the label tree to turn positive when any of its ancestors is negative. As all labels are known during training time, the running variables are always forced either below or above the threshold. By doing so, the regression coefficients $\eta_{l,k}$ are trained to reflect the interaction between the observed labels and the latent topics identified in the top part of the graphical model.

Note that the top part that identifies latent topics is not the actual aim of this model. The crucial aspect are the running variables and the label dummies: the latent topics are merely a tool to help achieve that goal.

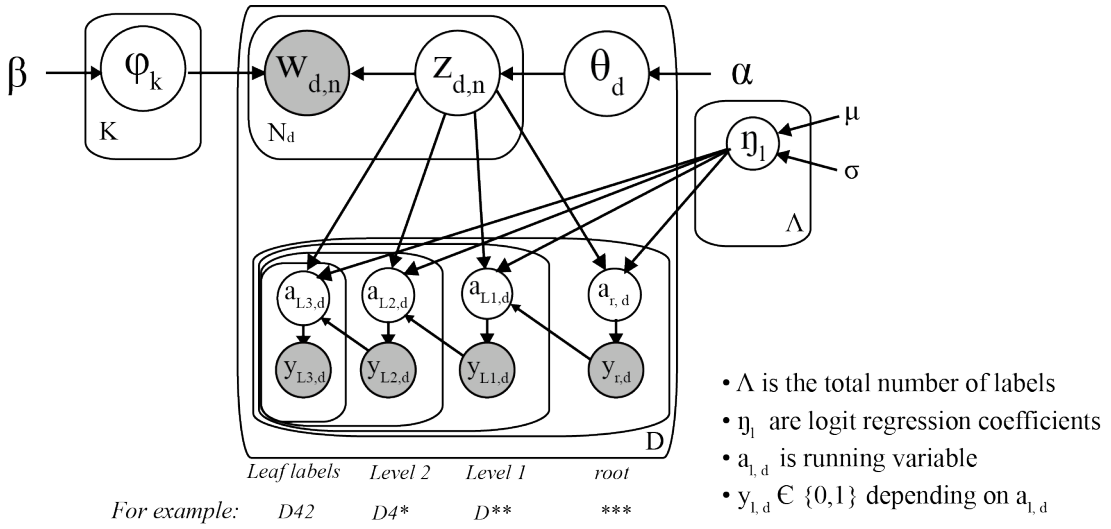

Figure 4: Graphical model: Hierarchically Supervised LDA

Except for the specification of α and β[3], this generative model corresponds to the original HSLDA:

1. For each topic $k = 1, \ldots, K$
   - Draw a distribution over words $\boldsymbol{\phi}_k | \boldsymbol{\beta} \sim \text{Dir}_V(\boldsymbol{\beta})$
2. For each label $l \in \mathcal{L}$
   - Draw a label application coefficient $\boldsymbol{\eta}_l | \boldsymbol{\mu}, \boldsymbol{\sigma} \sim \mathcal{N}_K(\boldsymbol{\mu}, \boldsymbol{\sigma})$, where $\boldsymbol{\mu}$ contains K identical values and $\boldsymbol{\sigma} = \text{diag}(\sigma)$ for $\sigma > 0$..
3. For each document $d = 1, \ldots, D$
   - Draw topic proportions $\boldsymbol{\theta}_d | \alpha \sim \text{Dir}_K(\alpha)$
   - For $n = 1, \ldots, N_d$
     - Draw a word assignment $z_{d,n} | \boldsymbol{\theta}_d \sim \text{Cat}_K(\boldsymbol{\theta}_d)$
     - Draw a word $w_{d,n} | z_{d,n}, \boldsymbol{\phi} \sim \text{Cat}_V(\boldsymbol{\phi}_{z_{d,n}})$
   - Set the root node $y_{root,d} = 1$
   - For each label l in $\mathcal{L}$, starting at the children of *root*

[3]The original prior structure uses Hierarchical Dirichlet Priors (HDP). Due to its complex construct and a lack of added value in our context, this aspect is excluded from the test. The Python code does feature HDP.

- Draw $a_{l,d}|\bar{\boldsymbol{z}}_d, \boldsymbol{\eta}_l, y_{pa(l),d} \sim \begin{cases} \mathcal{N}(\bar{\boldsymbol{z}}_d^T \boldsymbol{\eta}_l, 1), & \text{if } y_{pa(l),d} = 1 \\ \mathcal{N}(\bar{\boldsymbol{z}}_d^T \boldsymbol{\eta}_l, 1)\mathbb{1}_{\{a_{l,d}<0\}}, & \text{if } y_{pa(l),d} = -1 \end{cases}$
- Apply label l to document d according to

$$y_{l,d}|a_{l,d} = \begin{cases} 1 & \text{if } a_{l,d} > 0 \\ -1 & \text{else} \end{cases}$$

where $\bar{\boldsymbol{z}}_d^T = (\bar{z}_1, \ldots, \bar{z}_K)$ are the fractions of words that are assigned to topic k in document d, i.e. $\bar{z}_1 = \frac{1}{N_d} \sum_{n=1}^{N_d} \mathbb{1}_{\{z_{d,n}=k\}}$. This may be regarded as an intermediate estimate of $\boldsymbol{\theta}_d$.

The hierarchical coupling of the regression coefficients $\boldsymbol{\eta}_l$ creates a posteriori dependence which causes label predictors deeper in the hierarchy to focus on distinguishing features between a label's levels.

3.2.2 Gibbs Sampling

Collapsed Gibbs sampling is used for approximate inference in HSLDA. Whereas the baseline LDA model collapsed out all latent variables except $z_{d,n}$, HSLDA involves two additional latent conditional distributions, namely $a_{l,d}$ and $\boldsymbol{\eta}_l$.

- The conditional posterior distribution of the words' topic assignments $\mathbf{z_{d,n}}$ used in this paper differs in some important ways from the original HSLDA specifications. Instead of conditioning on the running variables $\mathbf{a}$, we will condition on $\mathbf{y}$. The reason for this is discussed a bit further below. This leads to

$$\begin{aligned} p(z_{d,n}|\mathbf{z}_{-d,n}, \mathbf{y}, \mathbf{w}, \boldsymbol{\eta}, \alpha, \beta) &= \frac{p(\mathbf{y}, z_{d,n}|\mathbf{z}_{-d,n}, \mathbf{w}, \boldsymbol{\eta}, \alpha, \beta) \cdot p(\mathbf{z}_{-d,n}, \mathbf{w}, \boldsymbol{\eta}, \alpha, \beta)}{p(\mathbf{z}_{-d,n}, \mathbf{y}, \mathbf{w}, \boldsymbol{\eta}, \alpha, \beta)} \\ &\propto p(\mathbf{y}_d, z_{d,n}|\mathbf{z}_{-d,n}, \mathbf{w}, \boldsymbol{\eta}, \alpha, \beta) \\ &= p(\mathbf{y}_d|z_{d,n}, \mathbf{z}_{-d,n}, \mathbf{w}, \boldsymbol{\eta}, \alpha, \beta) \cdot p(z_{d,n}|\mathbf{z}_{-d,n}, \mathbf{w}, \boldsymbol{\eta}, \alpha, \beta) \\ &= p(\mathbf{y}_d|\mathbf{z}, \boldsymbol{\eta}) \cdot p(z_{d,n}|\mathbf{z}_{-d,n}, \mathbf{w}, \alpha, \beta) \end{aligned}$$

where the conditioning sets of both terms could be reduced in the final line, because those nodes are outside the relevant Markov blanket (Lemma 2), and $z_{d,n}$ and $\boldsymbol{\eta}$ are d-separated by $a_{l,d}$ (Lemma 3). Considering that the variables in $\mathbf{y}_d$ can only take

values in $\{0, 1\}$, we know that

$$\begin{aligned} p(\mathbf{y}_d|\mathbf{z}, \boldsymbol{\eta}) &= \prod_{l \in \mathcal{L}_d} p(y_{l,d} = 1|\mathbf{z}, \boldsymbol{\eta})^{y_{l,d}} \cdot p(y_{l,d} = 0|\mathbf{z}, \boldsymbol{\eta})^{1-y_{l,d}} \\ &= \prod_{l \in \mathcal{L}_d} p(a_{l,d} > c|\mathbf{z}, \boldsymbol{\eta})^{y_{l,d}} \cdot p(a_{l,d} < c|\mathbf{z}, \boldsymbol{\eta})^{1-y_{l,d}} \\ &= \prod_{l \in \mathcal{L}_d} \int_c^{\infty} \exp\left\{-\frac{1}{2}(\bar{\mathbf{z}}_d^T \boldsymbol{\eta}_l - a_{l,d})^2\right\} \mathrm{d}a_{l,d} \quad \cdot \\ &\qquad \prod_{j \notin \mathcal{L}_d} \int_{-\infty}^{c} \exp\left\{-\frac{1}{2}(\bar{\mathbf{z}}_d^T \boldsymbol{\eta}_j - a_{j,d})^2\right\} \mathrm{d}a_{j,d} \end{aligned} \tag{19}$$

This derivation differs in two ways from the original HSLDA formulation. Firstly, the original HSLDA takes only the product for present labels into account, thus effectively ignoring the second term. This is justified by the assumption that absent labels may actually be part of the document's *true* label set as opposed to its *observed* label set, so the absent labels are not restricted to be below the threshold. This implies that the second set of integrals in (19) are evaluated from $-\infty$ to ∞. Since these integrals are proportional to probability density functions, they can be ignored. For the corpora in this thesis, however, it is implicitly assumed that the labels assigned to a document indicate that paper's true present (positive) labels, and therefore also its absent (negative) labels. Secondly, this paper focuses on $p(y_{l,d}|\mathbf{z}, \boldsymbol{\eta})$ (i.e. the conditional probability that $a_{l,d}$ is larger or smaller than threshold c) instead of $p(a_{l,d}|\mathbf{z}, \boldsymbol{\eta})$. The reason being that the former quantifies the distance from the threshold better than the latter and thus also captures the "magnitude of presence" more accurately.

The second term in (19) is equivalent to the results in Section 2.2 (Eq. 11):

$$p(z_{d,n}|\mathbf{z}_{-d,n}, \mathbf{w}, \alpha, \beta) = \frac{n_{-i,(k)}^{w_i} + \beta}{n_{-i,(k)} + V\beta} \cdot \frac{n_{-i,(d)}^{k} + \alpha}{n_{-i,(d)} + K\alpha}$$

- The conditional posterior distribution of the regression coefficients $\boldsymbol{\eta_l}$ corresponds to the least squares regression results with priors μ and σ conditional on the available data. In the statistics literature, this is referred to as a ridge regression:

$$p(\boldsymbol{\eta}_l|\mathbf{z}, \mathbf{a}, \mu, \sigma) = \mathcal{N}(\hat{\boldsymbol{\mu}}_l, \hat{\boldsymbol{\Sigma}})$$

where

$$\hat{\boldsymbol{\mu}} = \hat{\boldsymbol{\Sigma}}\left(\mathbf{1}\frac{\mu}{\sigma} + \bar{\mathbf{Z}}^T \mathbf{a}_l\right) \quad \text{and} \quad \hat{\boldsymbol{\Sigma}}^{-1} = \mathbf{I}_K \sigma^{-1} + \bar{\mathbf{Z}}^T \bar{\mathbf{Z}}$$

- The conditional posterior distribution of the running variables $a_{l,d}$ is closely related to the model specification for $a_{l,d} \mid \bar{\mathbf{z}}_d, \boldsymbol{\eta}_l, y_{pa(l),d}$. The difference is that the probability is

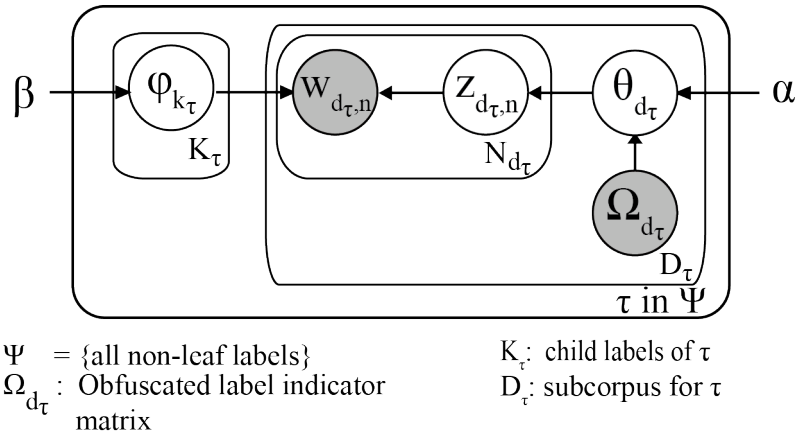

Figure 5: Graphical model: CascadeLDA

forced to zero (truncated) if the *is-a* hierarchy between $y_{l,d}$ and $a_{l,d}$ is violated. This results in

$$p(a_{l,d}|\mathbf{z}, \mathbf{Y}, \boldsymbol{\eta}) \propto \begin{cases} \exp\{-\frac{1}{2}(\bar{\mathbf{z}}_d^T \boldsymbol{\eta}_l - a_{l,d})^2\} \mathbb{1}_{\{a_{l,d} y_{l,d} > 0\}} & \text{if } y_{pa(l),d} = 1 \\ \exp\{-\frac{1}{2}(\bar{\mathbf{z}}_d^T \boldsymbol{\eta}_l - a_{l,d})^2\} \mathbb{1}_{\{a_{l,d} y_{l,d} > 0\}} \mathbb{1}_{\{a_{l,d} < 0\}} & \text{if } y_{pa(l),d} = -1 \end{cases}$$

3.3 Extension 3: CascadeLDA

CascadeLDA is designed to take advantage of the hierarchy structure in the labels with a different approach from the HSLDA mechanism. A foreseeable issue with the previously discussed extensions is that it seems unlikely for the topic-word distributions $\boldsymbol{\phi}_k$ to be discriminative between sibling labels. By focusing on the entire corpus -or global scope- the topic-word distributions of leaf labels are bound to be polluted by the same dominant words that dominate their siblings' (and parents') topic-word distributions. On the one hand, this does represent that leaf label's topic-word distribution, but on the other hand it does not help us in discriminating between sibling labels. CascadeLDA aims at finding discriminating features between sibling nodes by zooming in on a single (non-leaf) label at a time. The local scope then serves as a magnifying glass for the differences between sibling nodes.

Before proceeding, it is useful to be clear about the terminology used to explain CascadeLDA. Firstly, the terms sibling label and neighbour label may be used interchangeably. Secondly, leaf labels are not necessarily level 3 labels. As the name suggests, it refers to the most detailed labels in the label tree. If the model ignores the level 3 labels, then the level 2 automatically become the leaf labels. Thirdly, all labels with descendants (i.e. all non-leaf labels) can serve as parent labels. Similarly, all labels except the ROOT-label can serve as

child labels. Fourthly, each non-leaf label serves as the basis for a local scope. A local scope may also be referred to as neighbourhood or family scope.

The idea of CascadeLDA is to train an ensemble of L-LDA models in different local scopes. Each non-leaf label serves as a starting point of a local scope once during training time. This non-leaf label is called the parent label for that scope. All documents that have the parent label in their label sets are retained in the local scope, whereas all other documents fall outside the scope. For the retained documents the label sets are obfuscated to only contain the parent's children labels (i.e the labels that descent from the parent label in Figure 3). The remaining documents and labels are referred to as subcorpus. After creating the subcorpus, L-LDA is performed on it. As a final, but crucial, manipulation of the subcorpus, a generic label is added to each document's label set in addition to the children labels. This is the local scope analog of the ROOT-label.

Upon discarding all non-child labels, all non-family features of a document will be assigned to the generic label during training. In fact, all features shared by the documents in the subcorpus are assigned to the generic label. This includes generic parent label features. The only parts of a document that are likely to be attributed to the children labels are the ones that set a child label apart from its siblings.

After the L-LDA model in a local scope converged, the resulting topic-word distributions for the children labels are retained in the CascadeLDA model. Typically, CascadeLDA will then cascade down the label tree and choose one of the former child labels as the next parent label. This process is continued until all non-leaf labels have served as a parent label. After cascading down all branches, the CascadeLDA should have the topic-word distributions that distinguish sibling labels from each other.

In the classic L-LDA approach only leaf labels are considered labels. That is, no topic-word distribution is identified for e.g. the non-leaf label D8. In CascadeLDA, however, all non-leaf labels also have their own topic-word distributions. It may seem like a small detail, but leads to an important characteristic of CascadeLDA, because the topic-word distributions are *only* valid (i.e. usable) in their respective scope. A discriminating feature in a local scope may very well be completely uninformative in the global scope. To see why, consider a human family in which the children are recognised based on their hair colour. Even though this may help in their family home, it will not suffice to identify the children outside the family setting.

This creates a challenge during testing time: in which scope should an unseen document be tested? The process of testing starts at the root and attempts to assign the document to

one or more level 1 labels, in the same way L-LDA is applied for classification. After this, the testing procedure cascades down to each of the promising level 1 labels and the document's content is fitted to that parent label's local scope to discriminate between its children labels. This procedure is continued until all promising branches are pursued. A discussion on which labels are considered 'promising' is postponed to Section 5.1.

Beware of certain dynamics that should be handled with caution. Consider the situation in which the words for a test document are convincingly assigned to e.g. label B. As a result, the model will zoom in on label B and make it the parent in the following local scope. Here it may occur that the test document does not contain any features that allow for recognition of B's children, even though the document really is a member of the B-family. As a result, all words in the test document are assigned to the generic label and no further paths are considered in this branch and the prediction strands at a non-leaf label. For the datasets we are using this is not a favourable characteristic, because only leaf labels are assigned to documents.

4 Data and Preprocessing

Two corpora containing JEL-labeled, academic texts are at our disposal to which the methods discussed are applied.

The first corpus consists of 4,635 abstracts that each contain an average of 60 (standard deviation: 18.9) unique tokens. In order to obtain a wide range of topics, the abstracts were gathered from different journals including *American Economic Journals* with the specialisation journals *Applied Economics*, *Economic Policy*, *Macroeconomics* and *Microeconomics*. The creator of the JEL codes, the *Journal of Economic Literature*, is also the source of numerous abstracts in this research. Thus, the corpus should constitute a representative sample of the relevant academic economic literature of the past 20 years.

The other corpus is a collection of 1,422 full-text academic articles. Approximately half of the articles correspond to the same documents as the first corpus. The other half stems from the *Collaborative Research Center 649: Economic Risk* which is part of the interdisciplinary *Center for Applied Statistics and Economics*. All full-text articles have been gathered as PDF documents. After extracting the text from the documents all tables, images and non-text data has been removed.[4] The documents contain on average 5,444 unique words (standard

[4] The Python source code: `https://github.com/KenHBS/pdf_to_text`

	Abstracts until level 2	Abstracts until level 3	Full texts until level 2	Full texts until level 3
Nr. of documents	4,635	4,635	1,422	1,422
Nr. of words per doc.	59.8	59.8	5,444	5,444
Leaf labels only:				
- Nr. of labels	103	391	86	295
- Nr. of labels per doc.	2.9	3.6	3.0	3.5
- Distinct label sets	2,883	3,813	900	1.132
Incl. non-leaf labels:				
- Nr. of labels	122	512	103	398
- Nr. of labels per doc.	5.3	8.8	5.13	8.7

Table 3: Summary statistics of the corpora

deviation: 2,228).

After training the classification models, the next step is to test and evaluate its effectiveness. As is standard in classification problems, the model will be applied to unseen documents, also known as the test set. Even though labels are available for the test set - making them theoretically suitable to use for training - all documents in the test set have been isolated from the model during training. Failing to strictly split training data from test data is likely to lead to overfitting and the external validity of the classification would immediately be compromised.

The documents are labeled by humans. Authors have a subjective interpretation of the paper's semantic topics. This interpretation may or may not align well with the way other authors label their papers. This leads to inconsistent labelling of documents and is also known as inter-indexing inconsistency (Hamill and Zamora, 1980). It is useful to keep this potential flaw in the data quality in mind when interpreting the results of the model.

Labels that are used in less than 4 documents were removed from the datasets, due to a lack of data to create reliable predictions. In theory, 989 JEL labels exist (level 1, level 2 and leaf labels) of which 846 are leaf labels. To see what portion of the labels was retained after cleaning the labels please refer to Table 3.

All texts are transformed to bags-of-words and stemmed with the Porter Stemmer using the standard settings of the Python module gensim (Řehůřek and Sojka, 2010).

5 Evaluation Methods and Experiment Setup

In this section we will discuss the challenges faced during the task of classifying documents with the particular corpora at hand. Furthermore, the metrics that are used to evaluate the predictive quality of the models are introduced and discussed.

5.1 Challenges in Classification

To point out the challenges faced when predicting a document's JEL labels, an elaborate overview on the differences between the most basic classification methods and the LDA framework adopted in this paper will be given in this subsection. Additionally, it is important to stress the influence that classification methods and parameters (e.g. thresholds) may have on the quality of classifications.

The first difference between common use cases of classification models and this thesis is the number of classes in the problem. Elementary classifications focus on binary classification, that is, a model is trained to predict either zero or one (false or true) for a subject. Our setting is one of *multi-class classification* in which every JEL leaf label is considered its own class. Instead of predicting a zero or one, a model is trained to choose between K values, where K is the number of classes. Roughly two techniques exist to approach the multi-class problem: *one-vs-all* and *one-vs-one*. Multiple classifiers are needed in either case, because K classes need to be separated either way. The one-vs-all strategy creates a classifier for every class to separate itself from all other subjects. In order to predict the class of an unseen instance, every separating function is applied to that instance and thus it is assigned K values. The discriminating function that produces the highest score is chosen, because it supposedly separates itself the best from the rest. If a number of classes are more closely related than one distant class, the one-vs-all approach will tend to wrongfully predict the distant class. This is because the two related classes are hard to separate from each other and therefore they tend to rarely distinguish themselves from the rest. The one-vs-one approach does not suffer from this setting, because it creates a single classifier between every combination of two classes, resulting in $K(K-1)/2$ classifiers. The class that "wins" the most one-on-one comparisons will be the final prediction of a one-vs-one.

Secondly, a single instance may be assigned to more than one class. This is known as a *multi-label classification* problem. Methods dealing with such cases either transform the problem to a single-label problem or transform a single-label algorithm to match the multi-label requirement (Read et al., 2011; Tsoumakas and Katakis, 2007). A profound

transformation method is the *binary-relevance* method which applies a binary transformation to each label. That is, all labels are trained and predicted independent of each other, thus ignoring dependence structures between the labels. Selecting the threshold for including a label in the prediction requires more thought than in the single-label case, too. This threshold can either be defined in terms of a label's probability regardless of the other labels or all label probabilities are ranked and the N most probable labels are predicted. Surely, a hybrid of these two is also possible.

To put the above in perspective with the LDA extensions in this paper, note that HSLDA is a model designed for classification and produces proper probabilities, whereas L-LDA and CascadeLDA produce document-topic mixtures for each test document. Lets elaborate a bit: The fundamental aim of LDA is to learn topic-word distributions and document-topic mixtures. Establishing topic-word distributions could be considered the aim during training time, whereas the document-topic distribution is the important result during testing time. The outcomes of L-LDA and CascadeLDA are mixtures of topics which could be interpreted as the share of the document concerned with those topics. For a given document, $\sum_{k=1}^{K} \theta_{d,k} = 1$. With this in mind, also consider that multiple labels (eight maximum) may be applied to a single document. Now it becomes clear that even though a label may only account for 10 percent of the document mixture, this may still constitute a significant share in the document. On the other extreme, posterior inference on an unseen document may result in a 90 percent share of a given label and the remainder is allocated to another label. In such a case, a single label as the prediction seems more appropriate. Hence, the label's share in the document-topic mixture may not be interpreted as the probability of that label being present in the document. Therefore, it would be inappropriate to drive the decision of including a label in the prediction solely by the share in the document-topic mixture. A more suitable approach would incorporate a ranking of the document-topic mixtures and only include low share labels if they are the third or fourth most significant label. This is the approach used in this paper. The exact features of this hybrid prediction rule vary among the datasets and models.

As mentioned above, a significant weakness of the one-vs-all approach (see the paragraph on multi-classification) is that it favours classes farther away from other classes. If a number of classes are located very closely together, they are unlikely to be identified at all by a one-vs-all approach. This is where CascadeLDA comes in. Rather than focusing on all documents and classes, it takes a subset of both and aims to discriminate between the neighbouring classes (previously referred to as siblings). This still constitutes a one-vs-all approach, albeit

within a given local neighbourhood where distant neighbours are unlikely. CascadeLDA may be viewed as an ensemble of one-vs-all approaches. Creating such collections of multi-class subproblems is also referred to as hierarchical classification (Silva-Palacios et al., 2017).

5.2 Metrics for Classification Quality

A variety of metrics exist that help evaluate the quality of classification results. None of them captures the notion of classification quality perfectly. Therefore, each metric tends to focus on a different aspect of classification. It is thus very possible that model A outperforms model B according to one metric, but is inferior in terms of another metric. The multi-label nature adds complexity to the evaluation metrics and numerous approaches have been proposed (Tsoumakas and Katakis, 2007). A thorough comparison between the LDA extensions will therefore require an array of metrics for all corpora, label depths and models.

Before presenting the actual metrics it is appropriate to put the choice of metrics into context and be clear about what features they do and do not capture. To illustrate this, lets take a step back and focus on some of the LDA mechanisms that may cause inflated or deflated quality metrics. Recall that the Gibbs update for $z_{d,n}$ in Section 2.2 consists of two parts. One concerns the probability that word $w_{d,n}$ is part of topic k via the topic-word distribution, whereas the other term is related to document d's current document-topic mixture. This causes the update rule to favour currently present topics while updating $z_{d,n}$. One consequence of this is that small document-topic mixtures tend to vanish over the course of iterations to the benefit of already present topics. This is the root of two potential biases in classification metrics.

First of all, if the number of labels is large and some of the topic-word distributions are very similar, there is a risk of getting off on the wrong foot. As long as the topic-word distributions are nearly identical, there is no reason for the Gibbs sampler to move away from its current label in favour of its almost identical neighbour. As a result, LDA may often predict neighbours of a document's labels. In the setting of document classification, misclassifying a label with its neighbour should be regarded less of a misclassification than predicting a label from the other side of the spectrum. Due to the generative nature of LDA, such 'loss asymmetries' in misclassification are not implementable in a straightforward fashion if at all. For conventional machine learning methods, asymmetric loss functions can be implemented during training already thus affecting a model's actual classification ability (Abe et al., 2004; Zhou and Liu, 2006). For the LDA extensions in this paper, however,

the asymmetric costs of misclassification could be used to get a more accurate metric for classification quality, rather than actually affecting the predictions.

A second issue that originates from the generative mechanics of LDA is concerned with the majority of topics having zero document-topic mixtures[5]. As discussed before, numerous chains of events can cause a real label to be absent in the predictions. When all small share topics are shrunk to zero it is not uncommon for 90 percent of the labels to have a document-topic mixture share of zero. By ranking the predictions of the labels in a descending order of presence, the trailing 90 percent of the labels are randomly sorted. Rank based metrics that rely on the distance between the lowest ranked true label and highest ranked false label are therefore not useful for this research.

For each test document the K labels are ranked from most- to least likely. Based on this ranking the following *rank-based* evaluation metrics are calculated:

- AUC_{ROC}: The Area under the Curve (AUC) of the Receiver Operating Characteristic (ROC)-curve takes values between zero and one for every test document. Every point on an ROC-curve, (FPR(c_i), TPR(c_i)), is the False Positive Rate (FPR) vs. True Positive Rate (TPR) at a threshold c_i, where c_i moves from zero to one. $\text{FPR} = \frac{\text{FP}}{\text{FP + TN}}$ and $\text{TPR} = \frac{\text{TP}}{\text{TP+FN}}$, where FP is False Positive, TN is True Negative, TP is True Positive and FN is False Negative. At any given threshold c_i the labels will be given value true or false. FP, TN, TP and FN are the values in the cells of a contingency table at threshold c_i. The AUC_{ROC} therefore captures the accuracy in ranking. It may be interpreted as the probability that a randomly selected subject with label k is predicted to have a higher probability than a random subject without label k (Hanley and McNeil, 1982). When the ranking of likelihoods of absent labels and present labels are scattered randomly, the AUC_{ROC} will be 0.50, whereas a perfect prediction results in 1. Hence, the value of AUC_{ROC} should always be compared to 0.50. After obtaining an AUC_{ROC} for every test document, they are averaged to get the macro AUC_{ROC} as a single metric for a model's predictive capability:

$$\text{AUC}_{\text{ROC, macro}} = \frac{1}{D_{\text{test}}} \sum_{d=1}^{D_{\text{test}}} \text{AUC}_{\text{ROC},d}$$

- One-Hit: Percentage of documents for which the highest ranked label is correct.

[5]This is relevant only for L-LDA and CascadeLDA. In the case of HSLDA, $p(y_{l,d} = 1|\mathbf{z}, \boldsymbol{\eta})$ is used to classify documents, whereas for L-LDA and CascadeLDA the document-topic mixture is used as a basis for classification

- Two-Hit: Percentage of documents for which at least one of the two highest ranked labels is correct.

- F1-Score: Measure the harmonic mean between the TPR and 1-FPR. Since FPR may be considered as the probability of a false-alarm, either term is considered favourable regarding classification quality. F1-score is also measured regarding a threshold c_i. For this paper, the threshold that produces the highest F1-score for a given test document is chosen. This threshold is referred to with the subscript 'opt'.

$$\mathrm{F1}_{doc1} = \frac{2 \cdot \mathrm{TPR}_{opt} \cdot (1 - \mathrm{FPR}_{opt})}{\mathrm{TPR}_{opt} + (1 - \mathrm{FPR}_{opt})} \tag{20}$$

  The F1-scores of all test documents are then averaged to obtain the model's F1 score:

$$\mathrm{F1}_{\mathrm{macro}} = \frac{1}{D_{\mathrm{test}}} \sum_{d=1}^{D_{\mathrm{test}}} \mathrm{F1}_d$$

A word of caution is in place when interpreting and comparing the reported AUC_{ROC} metrics. By implementation, the L-LDA and CascadeLDA can obtain no predictions at all for a given document. Such documents are ignored while calculating evaluation metrics. To understand why this is done, we will inspect the cases in which the models produce no predictions. In the case of L-LDA this occurs when all word-topic assignment $z_{d,n}$ are assigned to the root label. As the root label is not a valuable prediction it is actively removed from the predictions as to not inflate the predictive quality for every single document. For CascadeLDA, the model may get stuck on its way down (see the end of Section 3.3). Because the evaluation metrics throw an error if all document-topic mixtures are zero, such documents are removed. All in all, CascadeLDA tends to end up with no solutions more often than L-LDA, especially in the academic abstract corpus. For the full text corpus, however, this rarely happens.

5.3 Experiment Setup

The quality of LDA and its extensions relies heavily on a number of settings. In this subsection the choices regarding hyperparameters and model settings are presented and discussed.

All models were estimated by means of collapsed Gibbs sampling (CGS). Even though more methods are available in theory, the only realistic competitor to CGS is CAVI. As pointed out by Asuncion et al. (2009), the claimed differences in performance between the different means of modelling are mainly attributable to hyper-parameter configurations. The choice for CGS was mainly driven by its relative simplicity and availability of rudimentary code examples.

The Gibbs sampler was programmed to run between 1,500 and 2,000 iterations in the L-LDA model. For reasons that will be discussed in Section 6.1, the CascadeLDA sampler produces the best results when less than 5 iterations per subcorpus are used.

The importance of hyperpriors has been pointed out in many papers (e.g Asuncion et al. (2009); Wallach et al. (2009)), but conclusive settings regarding the magnitude of the priors do not exist. After testing multiple magnitudes for the hyperpriors, the best results are achieved with α equal to 0.1 and β equal to 0.01. Without going into too much detail, it is notable that a larger β of, say, 0.1 actually produced better results in cases where the number of labels are small. Despite the lack of a highly detailed grid search, a considerable improvement was observed with relatively small changes in the hyperpriors.

To speed up the iterations, the documents are saved as (token-id, frequency) tuples. That is, instead of recording e.g. "market" a hundred times in a bag-of-words, it is recorded once, but is assigned a weight of hundred. Upon resampling, all instances of a word in a document are reassigned at the same time. This saves significant computational costs and was a necessary step to handle the full text corpus.

The least used words and the most used words were removed from the corpus. In addition to saving computational costs this should also remove noisy non-words that may have entered the files in the process of extracting text from the PDFs. Based on test runs, however, no pruning of the dictionary produced the best results in the corpus with abstracts. For the corpus with full texts, the results were unaffected by removing tokens that occur in less than 1 percent of the documents. Pruning the dictionary this way reduces the size of the vocabulary from 430,000 tokens to approximately 12,500. This resulted in a significant increase in speed.

After pruning the dictionary and saving the document as (token-id, frequency) tuples, the size of the full text corpus was reduced from nearly 7 million words to less than 1.4 million, without affecting predictive quality. Recall that every single word is assigned a topic in every single iteration. Hence, the effort to save computation time has actually resulted in a cost reduction of about 80 percent.

The number of topics K is determined by the number of unique labels in the corpus for L-LDA and CascadeLDA. In HSLDA, however, the number of topics K does not refer to the number of labels, but instead indicates the number of latent topics that should be used in running the probit regression on label presence. Upon personal inspection of the JEL codes a suitable number of semantic topics that were likely to be distinguishable was established to be 15.

By incorporating a Hierarchical Dirichlet Process (HDP), the hyperprior α was made flexible by estimating it based on the available data (Teh et al., 2006). The quality and stability of the latent topics are supposed to improve by this extension (Wallach et al., 2009). Unfortunately, the results for HSLDA were barely affected. Considering the trade-off between the complexity of the Hierarchical Dirichlet Process and its added value, it will not be covered in this paper.

6 Results

This section will present and discuss the classification quality of the models presented. The metrics will be based on the different LDA extensions, two datasets and multiple levels of detail in the labels. While comparing the models, much attention will be paid to the different settings in which each model thrives. This will expose the strengths of the models and identify the roots of their differences.

Despite promising features, some of the mechanics of HSLDA proved fatal for classifying academic articles according to JEL codes. The classification metrics are neither presentable nor interesting and have therefore been dismissed from this paper. Nevertheless, a detailed discussion on the reason for this outcome will be provided in Section 6.2, as it is certainly insightful.

6.1 L-LDA and CascadeLDA

Table 4 presents the evaluation metrics in multiple settings. No general consensus exists on acceptable values for AUC_{ROC}. However, values between 0.50 and 0.65 are usually considered poor, while values exceeding 0.90 are regarded as very good. Regardless of the interpretation it is safe to say the classifiers perform better than random predictions.

All evaluation metrics improve as the label depths decrease. Particularly the 1-Hit and 2-Hit metrics show a large increase. This is an expected result that was briefly touched upon in Section 5.2 when discussing the asymmetric costs of misclassifications. The notion that classifying sibling labels should not be penalised as much as classifying far-away labels can thus be implicitly incorporated by only considering labels up to level 2 in the label tree in Figure 3. A downside, however, is that a slightly misclassified document is now scored as a perfect prediction.

Another interesting finding is that neither model performed significantly better on the full text corpus. One may therefore conclude that the added number of words did not add features

Table 4: Evaluation metrics

Model	D*	Dataset	(i,t)**	AUC_{ROC}	1-Hit	2-Hit	F1-Score
L-LDA	3	Abs.	(2,000; 25)	0.862	0.655	0.779	0.536
L-LDA	2	Abs.	(2,000; 25)	0.885	0.748	0.838	0.643
L-LDA	1	Abs.	(2,000; 10)	0.898	0.817	0.929	0.789
CascadeLDA	3	Abs.	(4 ; 2)	0.901	0.573	0.699	0.518
CascadeLDA	2	Abs.	(4 ; 2)	0.934	0.748	0.860	0.680
CascadeLDA	1	Abs.	(4 ; 2)	0.941	0.846	0.951	0.857
L-LDA	3	Full	(1,500; 25)	0.818	0.524	0.727	0.514
L-LDA	2	Full	(2,000; 25)	0.849	0.678	0.846	0.644
L-LDA	1	Full	(1,500; 25)	0.922	0.916	0.979	0.863
CascadeLDA	3	Full	(2 ; 2)	0.884	0.490	0.671	0.512
CascadeLDA	2	Full	(2 ; 2)	0.906	0.678	0.804	0.656
CascadeLDA	1	Full	(2 ; 2)	0.906	0.825	0.944	0.820

Python code available at `https://github.com/KenHBS/LDA_thesis`

*: The label level (depth) considered in the analysis.

**: Parameters settings: i is the number of iterations in the Gibbs sampler, t is the number of iterations between consecutive saving of the Markov chain.

that help discriminate between labels. Both models apparently handle the (concentrated) information in the abstracts well enough to achieve high quality separation between labels.

6.1.1 Number of Iterations

A feature that requires a thorough discussion is the fact the CascadeLDA performs best with only 2 or 4 iterations of the Gibbs sampler. To explain this phenomenon, we need to answer two questions: Firstly, why does it produce good results after such a small number of iterations? Secondly, why does the performance not increase (in fact, it decreases) with more iterations?

The answer to the first question lies in the fact that CascadeLDA breaks a large multi-class problem down to multiple smaller multi-class problem. As the number of labels in each subcorpus is reduced drastically compared to the entire corpus, a stable state of the Gibbs sampler is expected to occur faster in each local scope. This expectation jumps to mind, because the number of topics/labels to choose from while assigning values to $z_{d,n}$ is relatively

small. As a result, the sampler in the subcorpus is more stable and settles in a steady state faster.

The second question can be tentatively answered by a mechanism we refer to as the *gravitational pull of the generic label*. Consider a document that contains four labels that belong to different branches of the label tree. By investigating this document in a subcorpus, three of those labels are obfuscated and only about 25 percent of that document's words are relevant to that local scope, because the other 75 percent belong to the obfuscated labels. The majority of that 75 percent is expected to be assigned to the generic label. Recall that the Gibbs update for the word-topic assignments favours topics that have a high document-topic loading already. This mechanism was also discussed in Section 5.2, albeit in a different setting. Over the course of iterations the generic label will be continuously favoured over the child labels, that are supposed to share the remaining 25 percent. As a result, word-topic assignments are gravitated towards the generic label and are gradually moved away from the child labels they actually belong to.

The two-fold answer describes two separate mechanisms that influence the state of the Markov chain simultaneously. Therefore, the argumentation that the Markov chain settles faster in a stable state only holds in absence of the gravitational pull of the generic label. In fact, the gravitational pull drags the Markov chain away from the optimal state and forces CascadeLDA to terminate with a small number of iterations. As a result, the predictive quality of CascadeLDA fluctuates heavily between runs.[6]

6.1.2 Speed Assessment

Now that we understand why less iterations are required in CascadeLDA, the next step is to assess what that means in terms of computational speed. The forces that increase CascadeLDA's computational costs compared to L-LDA will be presented first. Afterwards, the cost-saving mechanics are discussed. Finally, the advantage CascadeLDA has in terms of speed will be illustrated with a timing experiment.

Even though the number of iterations is small in a single subcorpus, there are approximately 125 subcorpora that are analysed. Additionally, a single document occurs in multiple local scopes. Even a document with only a single label occurs in three subcorpora: The first is headed by ROOT, and the second and third local scopes are headed by its level 1 and level 2 parents, respectively. Over the course of a single full iteration in CascadeLDA, a single

[6]The metrics presented in Table 4 correspond to the final runs of the model and are not necessarily the best values that were obtained.

document is analysed in at least three different local scopes.

The fewer number of iterations required are a tremendous push in terms of speed. Furthermore, in L-LDA every topic-word assignment involves calculating a K-dimensional vector of probabilities that is used to draw a value for $z_{d,n}$. In CascadeLDA, the number of possible labels, K_{local}, is much smaller and therefore computation is sped up. This effect is the largest if we considered all level 3 labels. Note, however, that even though the calculations of the K-dimensional vector of probabilities are vectorised with *numpy* (van der Walt et al., 2011), it still affects computational speed.

To avoid getting into a formal computational complexity analysis, a simple timing comparison was performed. The settings that resulted in the highest predictive quality for abstracts dataset were used for this assessment. The computation time reflects the time expired to load, train, test and evaluate the models with 4,635 abstracts. The results serve as a mere indication of the speed up and should not be interpreted as a formal statement. The timing comparison will vary depending on the machine used to compute, but also the dataset and the optimal settings are likely to influence the computation times of the models unequally. Having said that, the CascadeLDA requires 13 minutes to finish the task, whereas L-LDA ran for 300 minutes to attain a comparable predictive quality. Conservatively, CascadeLDA is faster than L-LDA by a factor of more than 20.

On a final note, CascadeLDA produces the results for all three label depths in a single run, whereas L-LDA requires training an entirely new model if the label depth is changed. The reason being that CascadeLDA is trained to perform in every local scope and thus also records topic-word distributions for non-leaf labels. L-LDA, on the other hand, only identifies topic-word distributions for leaf labels. If, for whatever reason, a classification at every label level is demanded, L-LDA will have to train three separate models.

6.2 HSLDA

HSLDA takes into account the hierarchical structure of the labels which was one of the main features that peaked interest in this model. L-LDA in particular fails to acknowledge this important piece of prior information. As a result, a suboptimal separation between sibling labels is obtained. The quality of HSLDA to overcome this remains theoretically convincing. The reason for the fatal HSLDA results lies in the structure of JEL codes.

Recall from Section 3.2 that a running variables $a_{l,d}$ is assigned to every document d for every label. $a_{l,d}$ follows a (truncated) normal distribution with mean $\bar{\mathbf{z}}_d^T \boldsymbol{\eta}_l$ and variance 1.

The mean of every running variable is thus a linear combination of a document's K latent topic mixtures and K regression coefficients. Simultaneously, $a_{l,d}$ is forced to be below or above the threshold, based on the presence of its parent label $pa(l)$ in document d's label set.

Upon integrating HSLDA in this thesis, the hope was that the labels down the hierarchy would be separated from each other by their running variable means $\bar{\mathbf{z}}_d^T \boldsymbol{\eta}_l$. Surely, within a single document the document-topic mixtures $\bar{\mathbf{z}}_d$ are equal for both labels l and l'. Separation of sibling labels l and l' would thus only occur if the regression coefficients $\boldsymbol{\eta}_l$ differs from the regression coefficient for label l'.

Note that the latent topics (whatever they may be) covered in sibling labels must be nearly identical for the JEL code labels, since they cover almost identical topics. As a result, the mechanism in HSLDA is unable to train discriminating regression coefficients.

At this point it may seem that HSLDA does not offer what it advertises. However, consider a catalog with product descriptions that are assigned to main categories and subcategories, such as the online catalog of Amazon.com. A single product may reside in multiple subcategories of different top-level product categories. Lets consider a typical product like sneakers. Sneakers appear in the top-level categories Sports & Outdoors and Clothing, Shoes & Jewelry. Within Sports & Outdoors the sneaker may also occur in Athletic Clothing as well as Sports Collectibles and so forth. In contrast to the JEL hierarchy, the sibling labels in e.g. Athletic Clothing are not very similar to sneakers and can therefore be discriminated very well. Taking into the account the prior information on all other products within a certain branch leads HSLDA to produce regression coefficients and latent topics that separate sibling labels from each other. Unfortunately, this has proven to not be the case for JEL code classification.

7 Conclusion

In conclusion, L-LDA and Cascade produce very comparable, high quality predictions in both datasets. The speed at which CascadeLDA attains these results is faster by a factor of roughly 20 to 25, but this is highly dependent on the label structure and general features of the dataset. The increase in speed comes at the cost of less stability. HSLDA could unfortunately not be included in the model comparison, due to the misalignment in label structure between our datasets and the datasets used in the original paper.

No systematic differences in predictive quality can be found between the corpora of academic abstracts and full texts. Apparently, all discriminative features contained in the full

texts are also present in the abstracts, as far as CascadeLDA and L-LDA are concerned.

7.1 Applications

The high quality predictions indicate that both L-LDA and CascadeLDA can be used to provide suggestions for JEL code labelings of unseen papers. A practical use case for either model would be to embed them in the uploading process of new academic articles. By analysing the abstract of the new article, the author can be quickly provided with the top 5 or 10 labels, including their descriptions. The authors would then be able to select the most fitting JEL codes by means of simple point-and-click. Such an implementation would have two benefits. Firstly, only relevant labels will be suggested based on the actual content of the paper (or abstract), leading to less subjective labelings. Secondly, the author would not have to go through the list of JEL codes and search for the most suitable codes for the paper. In the case of CascadeLDA, such a suggestion would take up less than half a second, whereas L-LDA would require approximately 10 seconds to perform the task.

CascadeLDA could be used in more general settings as well. Within the scope of text mining, it could be used to automatically archive new arrivals in a digital library. However, applications outside of text mining are also possible. Topic modelling techniques are used extensively in the field of bioinformatics for example to predict protein functions. As each protein may perform multiple functions and sub-functions (Cerri et al., 2016), these functions may be seen as labels and may be used the hierarchically predict the functions and sub-functions of unseen protein sequences.

7.2 Future Research

The final part of the thesis will be dedicated to ideas that may improve the models and are worth considering for future research.

An interesting concept to pursue is to prune the dictionary less crudely. The idea behind reducing the dictionary is to remove words and thus shrink the uninformative word share. Currently, words that occur in nearly all or barely any documents are removed from the corpus. A more refined way of doing this is to run an exploratory L-LDA and identify the top words for the generic label and remove them from the corpus. This is equivalent to cutting out the most frequent words. The least frequent and uninformative words could also be identified and cut away. To do so, the topic-word distribution matrix $\boldsymbol{\phi}$ of the exploratory L-LDA should be analysed. Recall that topic k represents a row $\boldsymbol{\phi}_{k,1:V}$. Instead of focusing

on the rows, one could analyse the columns and search for columns with a column sum close to zero. These columns correspond to words with low value in terms of identifying topics. By reducing the corpus in this way, the noise is removed from the corpus and the succeeding analysis will be more focused and less computationally expensive.

The idea of specialised pruning is particularly interesting for CascadeLDA for parent nodes further down the JEL label tree. The subcorpus contains only documents that belong (partially) to, say, label D8. In order to retain only relevant words for discriminating between siblings, a specialised pruning would remove all generic D8 terms *and* all terms that are irrelevant for D8. This will lead to a much smaller document-topic share of the 'rest'-label and would thus overcome the risk of the gravitational pull. Therefore, this may stabilise CascadeLDA's performance.

From a computer science point of view, it would be interesting to thoroughly analyse the computational complexity of the different models. This could formally validate the increase in speed and potentially confirm the scalability of CascadeLDA.

References

Abe, N., Zadrozny, B., and Langford, J. (2004). An iterative method for multi-class cost-sensitive learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 3–11, New York, NY, USA. ACM.

Andrzejewski, D., Zhu, X., and Craven, M. (2009). Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, pages 25–32.

Asuncion, A., Welling, M., Smyth, P., and Teh, Y. W. (2009). On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 27–34, Arlington, Virginia, United States. AUAI Press.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York Inc.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*.

Blei, D. M. and Lafferty, J. D. (2006). Correlated topic models. In *In Proceedings of the 23rd International Conference on Machine Learning*, pages 113–120. MIT Press.

Blei, D. M. and McAuliffe, J. D. (2007). Supervised topic models. *Neural Information Processing Systems*.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Cerri, R., Barros, R. C., P. L. F. de Carvalho, A. C., and Jin, Y. (2016). Reduction strategies for hierarchical multi-label classification in protein function prediction. *BMC Bioinformatics*, 17(1):373.

Geiger, D., Verma, T., and Pearl, J. (1990). Identifying independence in bayesian networks. *Networks*, 20(5):507–534.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

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

Hamill, K. A. and Zamora, A. (1980). The use of titles for automatic document classification. *Journal of the American Society for Information Science*, 31(6):396–402.

Hanley, J. A. and McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36.

Jagarlamudi, J., Daumé, III, H., and Udupa, R. (2012). Incorporating lexical priors into topic models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 204–213, Stroudsburg, PA, USA. Association for Computational Linguistics.

Joachims, T. (2002). *Learning to Classify Text Using Support Vector Machines – Methods, Theory, and Algorithms*. Kluwer/Springer.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Mao, X.-L., Ming, Z.-Y., Chua, T.-S., Li, S., Yan, H., and Li, X. (2012). Sshlda: A semi-supervised hierarchical topic model. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 800–809, Stroudsburg, PA, USA. Association for Computational Linguistics.

McCallum, A. and Nigam, K. (1998). A comparison of event models for naive bayes text classification. In *IN AAAI-98 WORKSHOP ON LEARNING FOR TEXT CATEGORIZATION*, pages 41–48. AAAI Press.

Nam, J., Kim, J., Mencía, E. L., Gurevych, I., and Fürnkranz, J. (2014). Large-scale multi-label text classificationâĂŤrevisiting neural networks. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 437–452. Springer.

Perotte, A. J., Wood, F., Elhadad, N., and Bartlett, N. (2011). Hierarchically supervised latent dirichlet allocation. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F.,

and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 24*, pages 2609–2617. Curran Associates, Inc.

Ramage, D., Hall, D., Nallapati, R., and Manning, C. D. (2009). Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 248–256. Association for Computational Linguistics.

Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2011). Classifier chains for multi-label classification. *Machine Learning*, 85(3):333.

Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. `http://is.muni.cz/publication/884893/en`.

Robert, C. P. and Casella, G. (2005). *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Rubin, T. N., Chambers, A., Smyth, P., and Steyvers, M. (2012). Statistical topic models for multi-label document classification. *Mach. Learn.*, 88(1-2):157–208.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47.

Silva-Palacios, D., Ferri, C., and RamÃŋrez-Quintana, M. J. (2017). Improving performance of multiclass classification by inducing class hierarchies. *Procedia Computer Science*, 108:1692 – 1701. International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Tsoumakas, G. and Katakis, I. (2007). Multi-label classification: An overview. *Int J Data Warehousing and Mining*, 2007:1–13.

van der Walt, S., Colbert, S. C., and Varoquaux, G. (2011). The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering*, 13(2):22–30.

Wallach, H. M., Mimno, D. M., and McCallum, A. (2009). Rethinking lda: Why priors matter. In Bengio, Y., Schuurmans, D., Lafferty, J. D., Williams, C. K. I., and Culotta, A.,

editors, *Advances in Neural Information Processing Systems 22*, pages 1973–1981. Curran Associates, Inc.

Wang, S., Chen, Z., Fei, G., Liu, B., and Emery, S. (2016). Targeted topic modeling for focused analysis. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1235–1244, New York, NY, USA. ACM.

Wood, J., Tan, P., Das, A., Wang, W., and Arnold, C. (2016). Source-lda: Enhancing probabilistic topic models using prior knowledge sources. *ArXiv e-prints*.

Zhou, Z.-H. and Liu, X.-Y. (2006). On multi-class cost-sensitive learning. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pages 567–572. AAAI Press.

A Variational Factors in the Exponential Family

Let the full conditional of y_m be a member of the exponential family, i.e.

$$p(y_m|\mathbf{x}, \mathbf{y}_{-m}) = h(y_m) \exp\left[\eta_m(\mathbf{x}, \mathbf{y}_{-m})T(y_m) - a(\eta_m(\mathbf{x}, \mathbf{y}_{-m}))\right] \tag{21}$$

where $h(y_m)$ is the base measure; $\eta_m(\mathbf{x}, \mathbf{y}_{-m})$ is the natural parameter, that always depends on the known parameters/conditioning set; $T(y_m)$ are the sufficient statistics; and $a(\cdot)$ is the log-normaliser

Using the general expression for exponential family members and the solution for $q_m^*(y_m)$ in (14)

$$\begin{aligned}
q_m^*(y_m) &\propto \exp\left[\mathrm{E}_{-m}(\log p(y_m|\mathbf{x}, \mathbf{y}_{-m}))\right] \\
&= \exp\left[\mathrm{E}_{-m}\left(\log\{h(y_m) \exp\left[\eta_m(\mathbf{x}, \mathbf{y}_{-m})T(y_m) - a(\eta_m(\mathbf{x}, \mathbf{y}_{-m}))\right]\}\right)\right] \\
&= \exp \mathrm{E}_{-m}\left(\log\{h(y_m)\} + \eta_m(\mathbf{x}, \mathbf{y}_{-m})T(y_m) - a(\eta_m(\mathbf{x}, \mathbf{y}_{-m}))\right) \\
&= h(y_m) \exp\left[\mathrm{E}_{-m}\{\eta_m(\mathbf{x}, \mathbf{y}_{-m})\}T(y_m) - \mathrm{E}_{-m}\{a(\eta_m(\mathbf{x}, \mathbf{y}_{-m}))\}\right] \\
&\propto h(y_m) \exp\left[\mathrm{E}_{-m}\{\eta_m(\mathbf{x}, \mathbf{y}_{-m})\}T(y_m)\right]
\end{aligned}$$

The log-normaliser $a(\eta_m(\mathbf{x}, \mathbf{y}_{-m}))$ is the normalising factor that ensures that the exponential family member's density integrates to one. Therefore, it may be ignored, because it acts as a scaling constant in this optimisation problem.

The final expression reveals a crucial feature: the base measure $h(y_m)$ and sufficient statistic $T(y_m)$ are the same for the variational factor $q^*(y_m)$ as they are for the full conditional. This fact implies that the variational factor must be a member of the same exponential family of distribution as the full conditional. The difference between the variational factor and the full conditional lies *only* in the value of its natural parameters. In the variational factor $q^*(y_m)$, the natural parameter is equal to $\mathrm{E}_{-m}\{\eta_m(\mathbf{x}, \mathbf{y}_{-m})\}$ instead of $\eta_m(\mathbf{x}, \mathbf{y}_{-m})$.