# Latent Dirichlet Allocation and its application on a real-world dataset

Thomas Porter*
Department of Mathematical Sciences, University of Durham, UK
Email: mkhj24@durham.ac.uk

September 2, 2022

**Abstract**

This report provides an introduction to the latent Dirichlet allocation (LDA) model - a commonly used generative topic model which allocates a set of topics to a collection of documents by assuming each document exhibits each topic to different extents and each topic exhibits each word in the corpus to different extents. We discuss stochastic variational inference as a method of posterior inference for LDA. This combines stochastic optimisation and variational inference in order to approximate the posterior distribution of the latent variables given the observations in the LDA model. We then fit an LDA model to a dataset generated by scraping the text from 8422 BBC news articles, evaluating its performance using a series of visualisations of the resulting topics.

*Mr Thomas Porter is a MSc candidate from the University of Durham. This is a MSc dissertation submitted as part of the program MSc Scientific Computing and Data Analysis (G5K609) at the University of Durham. It was produced on September 2, 2022. It is supervised by Dr. Georgios Karagiannis from the Department of Mathematical Sciences in the University of Durham, UK.

# 1 Introduction

## 1.1 Topic models

Topic modelling is an unsupervised learning technique used to identify topics within a set of documents by analysing the words observed in each document. Topic models make use of the fact that documents tend to exhibit several topics, each in their own proportion. They form topics which are collections of similar words, again each with their own proportions within the topic. It is a very useful technique for the exploration and organisation of data, particularly for large collections of data that would take too long to organise manually. These collections are commonplace in modern days thanks to the growth of the internet which is estimated to run into the zettabytes ($10^{21}$ bytes). As well as text-mining, topic models are also used in fields such as computer vision (Cao & Fei-Fei 2007) and bioinformatics (Blei 2012). One of the earliest topic models was latent semantic indexing (LSI) (Papadimitriou et al. 2000) which uses spectral analysis of the term-document matrix - a matrix which stores the frequency of each term in a document - to capture the underlying semantics of a corpus. As one would expect when handling large collections of documents, the datasets that topic models analyse can often be huge and so the optimal topic models tend to be those that are both accurate and efficient. Luckily we have an ever-increasing amount of computational power at our fingertips and so we are able to use topic models on millions of documents at a time.

## 1.2 Aims of report

The aims of this report are threefold. Section 1 details a type of topic model known as latent Dirichlet allocation (LDA). This models a document as a distribution over the topics and models the topics themselves as a distribution over the words in the collection of documents. Section 2 discusses stochastic variational inference, the method used to approximate the

posterior distribution of the latent variables given the observations in the LDA model. Lastly, Section 3 applies LDA to a dataset that we have generated using Python. It consists of 8422 news articles with varying word counts and topics which we hope to capture with our topic model. We want to be able to draw insights from the model to get a better understanding of the types of topics exhibited by the dataset. We also want to gain an understanding of some of the similarities and differences between topics.

# 2 Latent Dirichlet allocation model

## 2.1 Background

Latent Dirichlet allocation or *LDA* ((Blei et al. 2003)), is a generative statistical model which assumes that within a set of documents or *corpus*, each document is distributed as a random mixture over latent topics and these topics are distributed according to a mixture over all of the words in the corpus. Each document consists of observations (in this case words) which are assumed to be drawn from a single topic. Please note that Sections 2, 3 and 4 follow the original paper for stochastic variational inference (Hoffman et al. 2012).

## 2.2 Notation

The notation we will use is displayed in Table 1. It is important to note that we will represent the categorical variables, $z_{d,n}$ and $w_{d,n}$ using indicator vectors. This is a vector of zeros with a single one, corresponding to the topic or word respectively. For example $z_{d,n}$ is a $K$-vector which represents the topic of the $n^{\text{th}}$ word in the $d^{\text{th}}$ document. Therefore, if $z_{d,n}^k = 1$ then this word has been assigned to the $k^{\text{th}}$ topic.

The dependencies between these variables can be represented in plate notation as illustrated in Figure 1. The shaded nodes represent observed variables whilst unshaded nodes indicate latent variables. Therefore, only the words, $w_{d,n}$ are observed. The topics, $\beta_k$, topic propor-

| Variable | Definition |
|---|---|
| $D$ | No. of documents in corpus |
| $K$ | No. of topics in corpus |
| $N_{d=1\dots D}$ | No. of words in document $d$ |
| $N$ | Total no. of words in corpus, i.e. $N = \sum_{d=1}^{D} N_d$ |
| $\alpha_{k=1\dots K}$ | Prior weight of topic $k$ in a document. The value of $\alpha_k$ is constant, i.e. $\alpha_k = \alpha \; \forall \; k$ |
| $\boldsymbol{\alpha}$ | Vector of prior weights $\alpha_k$ for $k = 1 \dots K$ |
| $\eta_{w=1\dots V}$ | Prior weight of word $w$ in a topic. The value of $\eta_w$ is constant, i.e. $\eta_w = \eta \; \forall \; w$ |
| $\boldsymbol{\eta}$ | Vector of prior weights $\beta_w$ for $w = 1 \dots V$ |
| $\beta_{k=1\dots K, w=1\dots V}$ | Probability of word $w$ being found in topic $k$ |
| $\boldsymbol{\beta}_{k=1\dots K}$ | Distribution of words in topic $k$ |
| $\theta_{d=1\dots D, k=1\dots K}$ | Probability of topic $k$ being found in document $d$ |
| $\boldsymbol{\theta}_{d=1\dots D}$ | Distribution of topics in document $d$ |
| $z_{d=1\dots D, n=1\dots N_d}$ | Topic of word $n$ in document $d$ |
| $\boldsymbol{z}_{d=1\dots D}$ | Topic of all words in corpus |
| $w_{d=1\dots D, n=1\dots N_d}$ | Word $n$ in document $d$ |
| $\boldsymbol{w}_{d=1\dots D}$ | All words in corpus |

Table 1: Description of variables in LDA

tions, $\theta_d$ and topic assignments, $z_{d,n}$ are all latent variables. The boxes or *plates* correspond to repeated entries. The larger plate represents documents while the smaller plate inside this corresponds to the word positions within each document. Lastly, the smallest plate represents the topics.

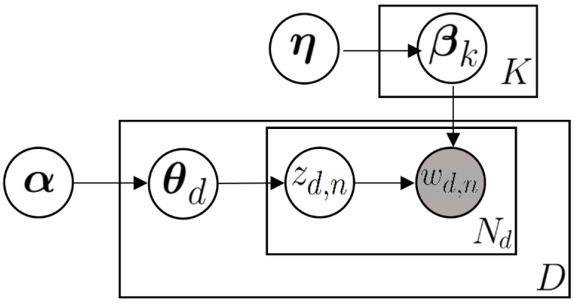


Figure 1: Plate notation for Latent Dirichlet allocation.

## 2.3 Model

The main assumption of LDA is that each document displays all of the $K$ topics to different extents and each topic displays all of the words in the corpus to different extents. It is a generative model which means it models the distribution of both the inputs and outputs. This means that sampling from the model allows the generation of synthetic data points from

the input space (Bishop 2006). The generative process of LDA is described in Algorithm 1 (Hoffman et al. 2012):

**Algorithm 1:** Generative process of latent Dirichlet allocation

**for** $k \in 1, ..., K$ **do**
  *Draw a topic distribution* $\boldsymbol{\beta}_k \sim Dirichlet(\eta, ..., \eta)$
**for** $d \in 1, ..., D$ **do**
  *Draw topic proportions* $\boldsymbol{\theta}_d \sim Dirichlet(\alpha, ..., \alpha)$
  **for** $n \in 1, ..., N_d$ **do**
    *Draw a topic assignment* $z_{d,n} \sim Multinomial(\boldsymbol{\theta}_d)$
    *Draw a word* $w_{d,n} \sim Multinomial(\boldsymbol{\beta}_{z_{d,n}})$

As we can see, the model assumes that both the topics $\boldsymbol{\beta}_k$ and topic proportions $\boldsymbol{\theta}_d$ have a Dirichlet prior. The Dirichlet distribution is a continuous probability distribution parameterised by a vector of positive real numbers. It is a multivariate generalisation of the beta distribution (Kotz et al. 2000). A $K$-dimensional Dirichlet distribution parameterised by $\boldsymbol{\gamma}$ has probability density function,

$$\text{Dirichlet}(\boldsymbol{\theta}; \boldsymbol{\gamma}) = \frac{\Gamma\left(\sum_{i=1}^{K} \gamma_i\right)}{\prod_{i=1}^{K} \Gamma(\gamma_i)} \prod_{i=1}^{K} \theta_i^{\gamma_i - 1} \tag{1}$$

where $\{\theta_i\}_{i=1}^{K}$ are on the $K-1$ simplex, i.e. $\sum_{i=1}^{K} \theta_i = 1$ and $\theta_i \geq 0$ for all $i \in \{1, \ldots, K\}$ and $\Gamma()$ represents the Gamma function. We have assumed that the parameters $\boldsymbol{\eta}$ and $\boldsymbol{\alpha}$ within each Dirichlet prior are constant, i.e. $\eta_w = \eta \; \forall \; w$ and $\alpha_k = \alpha \; \forall \; k$. This is known as an exchangeable prior.

The aim of LDA is to be able to find the posterior distribution $p(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z} \mid \boldsymbol{w})$ where $\boldsymbol{\beta} = \boldsymbol{\beta}_{1:K}, \boldsymbol{\theta} = \boldsymbol{\theta}_{1:D}, \boldsymbol{z} = \boldsymbol{z}_{1:D,1:N_d}$ and $\boldsymbol{w} = \boldsymbol{w}_{1:D}$. However, computing this distribution is an intractable problem. Luckily there exist a number of methods for approximating the posterior. Methods such as expectation propagation (Minka & Lafferty 2002), variational inference (Blei et al. 2003), Markov chain Monte Carlo methods (Griffiths & Steyvers 2004)

and likelihood maximisation (Alexander et al. 2009) can all be used to find an approximation of the posterior distribution. In this report we will use stochastic variational inference (Hoffman et al. 2012). This applies stochastic optimisation (Robbins & Monro 1951) to variational inference. Stochastic optimisation involves the minimisation or maximisation of an objective function in which randomness is present. When applied to variational inference, this results in an algorithm which iterates between taking a subsample from the dataset and re-estimating the latent structure of the model based on this subsample.

# 3 Stochastic variational inference

## 3.1 Class of models for variational inference

For ease of explanation we will define some new notation here so that we can learn how stochastic variational inference works for the general case. The class of models to which stochastic variational inference applies consists of observations, global latent variables, local latent variables and fixed parameters (Hoffman et al. 2012). Assume that we have a set of observations $\boldsymbol{y} = y_{1,N}$, a vector of global latent variables $\boldsymbol{\xi}$, a set of local latent variables $\boldsymbol{v} = \boldsymbol{v}_{1,N}$ where $\boldsymbol{v}_n = v_{n,1:J}$ and a vector of fixed parameters $\boldsymbol{\omega}$. For simplicity, assume that these fixed parameters only influence the global latent variables. The ultimate aim of stochastic variational inference is to be able to estimate the posterior distribution of the latent variables conditioned on our observations, $p(\boldsymbol{v}, \boldsymbol{\xi} \mid \boldsymbol{y})$. We can begin by noticing that the joint distribution of the latent local and global variables and observed variables conditioned on the fixed parameters can be factorised as follows,

$$p(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\xi} \mid \boldsymbol{\omega}) = p(\boldsymbol{\xi} \mid \boldsymbol{\omega}) \prod_{n=1}^{N} p(y_n, \boldsymbol{v}_n \mid \boldsymbol{\xi}). \tag{2}$$

We have that given the global variables $\boldsymbol{\xi}$ and fixed parameters $\boldsymbol{\omega}$, the n$^{th}$ setting of the local variables $\{y_n, \boldsymbol{v}_n\}$ is independent of all of the other observations and local variables (Hoffman et al. 2012). This can be written as,

$$p(y_n, \boldsymbol{v}_n \mid y_{-n}, \boldsymbol{v}_{-n}, \boldsymbol{\xi}, \boldsymbol{\omega}) = p(y_n, \boldsymbol{v}_n \mid \boldsymbol{\xi}, \boldsymbol{\omega})$$

where we have made use of the common notation $y_{-n}$ to denote the set of observations without $y_n$.

A complete conditional is the posterior distribution of a latent variable conditioned on all other variables in the model. We shall make the assumption that our complete conditionals are members of the exponential family,

$$p(\boldsymbol{\xi} \mid \boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega}) = h(\boldsymbol{\xi})\exp\{\boldsymbol{\eta}_g(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega})^T t(\boldsymbol{\xi}) - a_g(\boldsymbol{\eta}_g(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega}))\} \tag{3}$$

$$p(v_{n,j} \mid y_n, v_{n,-j}, \boldsymbol{\xi}) = h(v_{n,j})\exp\{\boldsymbol{\eta}_l(y_n, v_{n,-j}, \boldsymbol{\xi})^T t(v_{n,j}) - a_l(\boldsymbol{\eta}_l(y_n, v_{n,-j}, \boldsymbol{\xi}))\}. \tag{4}$$

The *base measure* $h(\cdot)$ and *log-normaliser* $a(\cdot)$ are both scalar functions whilst the *natural parameter* $\boldsymbol{\eta}(\cdot)$ and *sufficient statistics* $t(\cdot)$ are both vector functions. Since both complete conditionals belong to the same family, the global variables $\boldsymbol{\xi}$ and local variables $(y_n, \boldsymbol{v_n})$ are a conjugate pair. This implies that conditioned on the global variable, the distribution of the local variables is also in the exponential family,

$$p(y_n, \boldsymbol{v}_n \mid \boldsymbol{\xi}) = h(y_n, \boldsymbol{v}_n)\exp\{\boldsymbol{\xi}^T t(y_n, \boldsymbol{v}_n) - a_l(\boldsymbol{\xi})\}. \tag{5}$$

We must also have that the prior distribution $p(\boldsymbol{\xi})$ is a member of the exponential family (Hoffman et al. 2012),

$$p(\boldsymbol{\xi}) = h(\boldsymbol{\xi})\exp\{\boldsymbol{\omega}^T t(\boldsymbol{\xi}) - a_g(\boldsymbol{\omega})\}. \tag{6}$$

Since the sufficient statistics are $t(\boldsymbol{\xi}) = (\boldsymbol{\xi}, -a_l(\boldsymbol{\xi}))$, the hyperparameter $\boldsymbol{\omega}$ must be 2-dimensional; $\boldsymbol{\omega} = (\boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$. Equations 5 and 6 imply that $p(\boldsymbol{\xi} \mid \boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega})$ is a member of the same exponential family as the prior with natural parameter

$$\boldsymbol{\eta}_g(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega}) = (\boldsymbol{\omega}_1 + \sum_{n=1}^{N} t(\boldsymbol{v}_n, y_n), \boldsymbol{\omega}_2 + N). \tag{7}$$

As we said earlier, the ultimate aim of stochastic variational inference is to be able to estimate the posterior distribution of the latent variables conditioned on the observations. This can be written as

$$p(\boldsymbol{v}, \boldsymbol{\xi} \mid \boldsymbol{y}) = \frac{p(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\xi})}{\int p(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\xi}) d\boldsymbol{v} d\boldsymbol{\xi}}. \tag{8}$$

Unfortunately, computing the denominator in Equation 8 can often be intractable hence we need to use approximate posterior inference.

## 3.2 Mean-field variational inference

Variational inference involves finding a family of distributions over the latent variables which is indexed by a set of free parameters. The aim is to find the optimum of these parameters such that the distribution is closest in terms of Kullback-Liebler divergence to our posterior. The final distribution is known as the variational distribution. One particular instance of variational inference is mean-field variational inference. This differs from other forms of variational inference in that we make the assumption that each latent variable is independent. In order to minimise the Kullback-Liebler divergence between our variational distribution and the posterior, we maximise the *evidence lower bound* (ELBO). This is a lower bound on log $p(y)$ and equals the negative KL divergence up to a constant. In order to derive the ELBO we need to introduce a distribution over the latent variables $q(\boldsymbol{v}, \boldsymbol{\xi})$ and state Jensen's inequality (Jensen 1906). This states that for two sets of real numbers $\{\mu_1, ..., \mu_K\}$,

$\{r_1, ..., r_K\}$ with $\mu_k \geq 0$ and $\sum_{k=1}^{K} \mu_k = 1$ and a concave function $f$:

$$f\left(\sum_{k=1}^{K} \mu_k r_k\right) \geq \sum_{k=1}^{K} \mu_k f\left(r_k\right). \tag{9}$$

Since the logarithm function is concave we have that $\log \mathbb{E}[f(y)] \geq \mathbb{E}[\log f(y)]$. We hence have the following bound of the log marginal,

$$\begin{aligned}
\log p(y) &= \log \int p(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\xi}) d\boldsymbol{v} d\boldsymbol{\xi} \\
&= \log \int p(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\xi}) \frac{q(\boldsymbol{v}, \boldsymbol{\xi})}{q(\boldsymbol{v}, \boldsymbol{\xi})} d\boldsymbol{v} d\boldsymbol{\xi} \\
&= \log \left( \mathbb{E}_q \left[ \frac{p(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\xi})}{q(\boldsymbol{v}, \boldsymbol{\xi})} \right] \right) \\
&\geq \mathbb{E}_q[\log p(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\xi})] - \mathbb{E}_q[\log q(\boldsymbol{v}, \boldsymbol{\xi})] \\
&\triangleq \mathcal{F}(q)
\end{aligned} \tag{10}$$

Equation 10 consists of two terms; the expected log joint distribution, $\mathbb{E}_q[\log p(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\xi})]$ and the entropy of the variational distribution, $-\mathbb{E}_q[\log q(\boldsymbol{v}, \boldsymbol{\xi})]$. Notice that both of these terms are dependent on the variational distribution $q(\boldsymbol{v}, \boldsymbol{\xi})$ which we require to be in a tractable family of distributions so that the expectations can be computed easily. Once we have found such a family, the next task is to find the distribution within this family that maximises the ELBO. This is the same as finding the $q(\boldsymbol{v}, \boldsymbol{\xi})$ with the smallest KL divergence to the posterior (Jordan et al. 1999),

$$\begin{aligned}
\mathrm{KL}(q(\boldsymbol{v}, \boldsymbol{\xi}) \mid p(\boldsymbol{v}, \boldsymbol{\xi} \mid \boldsymbol{y})) &= \mathbb{E}_q[\log q(\boldsymbol{v}, \boldsymbol{\xi})] - \mathbb{E}_q[\log p(\boldsymbol{v}, \boldsymbol{\xi} \mid \boldsymbol{y})] \\
&= \mathbb{E}_q[\log q(\boldsymbol{v}, \boldsymbol{\xi})] - \mathbb{E}_q[\log p(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\xi})] + \log p(\boldsymbol{y}) \\
&= \mathcal{F}(q) + \text{const.}
\end{aligned}$$

The variational family of distributions that we will be using is the *mean-field family* in which

each latent variable is independent and has its own parameter (Hoffman et al. 2012),

$$q(\boldsymbol{v}, \boldsymbol{\xi}) = q(\boldsymbol{\xi} \mid \boldsymbol{\pi}) \prod_{n=1}^{N} \prod_{j=1}^{J} q(v_{n,j} \mid \sigma_{n,j}) \tag{11}$$

The variational parameters $\boldsymbol{\pi}$ and $\boldsymbol{\sigma}_n$ govern the global and $n^{\text{th}}$ local variables respectively. We require that $q(\boldsymbol{\xi} \mid \boldsymbol{\pi})$ and $q(v_{n,j} \mid \sigma_{n,j})$ are in the same exponential family as the complete conditionals $p(\boldsymbol{\xi} \mid \boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega})$ and $p(v_{n,j} \mid y_n, v_{n,-j}, \boldsymbol{\xi})$ in Equations 3 and 4,

$$q(\boldsymbol{\xi} \mid \boldsymbol{\pi}) = h(\boldsymbol{\xi}) \exp\{\boldsymbol{\pi}^T t(\boldsymbol{\xi}) - a_g(\boldsymbol{\pi})\} \tag{12}$$

$$q(v_{n,j} \mid \sigma_{n,j}) = h(v_{n,j}) \exp\{\sigma_{n,j}^T t(v_{n,j}) - a_l(\sigma_{n,j})\} \tag{13}$$

The benefits to these forms of the variational distributions are twofold. Firstly, they make the coordinate ascent algorithm more efficient. Secondly, the optimal mean-field distribution, regardless of its exact form, has factors in these families (Bishop 2006). For simplicity we will sometimes ignore the dependence on the variational parameters in our notation. One of the reasons we have chosen the mean-field family is that its entropy term decomposes,

$$\begin{aligned} -\mathbb{E}_q[\log q(\boldsymbol{v}, \boldsymbol{\xi})] &= -\mathbb{E}_q \left[ \log q(\boldsymbol{\xi} \mid \boldsymbol{\pi}) \prod_{n=1}^{N} \prod_{j=1}^{J} q(v_{n,j} \mid \sigma_{n,j}) \right] \\ &= -\mathbb{E}_{\boldsymbol{\pi}}[\log q(\boldsymbol{\xi})] - \sum_{n=1}^{N} \sum_{j=1}^{J} \mathbb{E}_{\sigma_{n,j}}[\log q(v_{n,j})] \end{aligned} \tag{14}$$

where $\mathbb{E}_{\sigma_{n,j}}[\cdot]$ and $\mathbb{E}_{\boldsymbol{\pi}}[\cdot]$ represent expectations with respect to $q(v_{n,j} \mid \sigma_{n,j})$ and $q(\boldsymbol{\xi} \mid \boldsymbol{\pi})$ respectively.

We will now derive the gradient of the ELBO before discussing coordinate ascent inference. Our goal is to optimise the gradients of the objective function from Equation 10 with respect to the variational parameters. We first derive the coordinate update for the global parameters

$\boldsymbol{\pi}$. The objective function can be written as a function of $\boldsymbol{\pi}$,

$$\mathcal{F}(\boldsymbol{\pi}) = \mathbb{E}_q[\log p(\boldsymbol{\xi} \mid \boldsymbol{y}, \boldsymbol{v})] - \mathbb{E}_q[\log q(\boldsymbol{\xi})] + c \tag{15}$$

where $c$ is constant with respect to $\boldsymbol{\pi}$. The first term uses the following derivation from $\mathbb{E}_q[\log p(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\xi})]$,

$$\begin{aligned}
\mathbb{E}_q[\log p(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\xi})] &= \mathbb{E}_q\left[\log[p(\boldsymbol{\xi} \mid \boldsymbol{y}, \boldsymbol{v})p(\boldsymbol{y}, \boldsymbol{v})]\right] \\
&= \mathbb{E}_q\left[\log p(\boldsymbol{\xi} \mid \boldsymbol{y}, \boldsymbol{v}) + \log p(\boldsymbol{y}, \boldsymbol{v})\right] \\
&= \mathbb{E}_q[\log p(\boldsymbol{\xi} \mid \boldsymbol{y}, \boldsymbol{v})] + \mathbb{E}_q[\log p(\boldsymbol{y}, \boldsymbol{v})].
\end{aligned}$$

Here the latter term is absorbed by the constant due to its independence from $\boldsymbol{\pi}$. Next we can substitute for $q(\boldsymbol{\xi} \mid \boldsymbol{\pi})$ from Equation 12 and $p(\boldsymbol{\xi} \mid \boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega})$ from Equation 3 into Equation 15,

$$\begin{aligned}
\mathcal{F}(\boldsymbol{\pi}) &= \mathbb{E}_q[\log[h(\boldsymbol{\xi})\exp\{\boldsymbol{\eta}_g(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega})^T t(\boldsymbol{\xi}) - a_g(\boldsymbol{\eta}_g(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega}))\}]] - \mathbb{E}_q[\log h(\boldsymbol{\xi})\exp\{\boldsymbol{\pi}^T t(\boldsymbol{\xi}) - a_g(\boldsymbol{\pi})\}] + c \\
&= \mathbb{E}_q[\log h(\boldsymbol{\xi}) + \boldsymbol{\eta}_g(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega})^T t(\boldsymbol{\xi}) - a_g(\boldsymbol{\eta}_g(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega}))] - \mathbb{E}_q[\log h(\boldsymbol{\xi}) + \boldsymbol{\pi}^T t(\boldsymbol{\xi}) - a_g(\boldsymbol{\pi})] + c \\
&= \mathbb{E}_q[\boldsymbol{\eta}_g(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega})]^T \nabla_{\boldsymbol{\pi}} a_g(\boldsymbol{\pi}) - \mathbb{E}_q[a_g(\boldsymbol{\eta}_g(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega}))] - \boldsymbol{\pi}^T \nabla_{\boldsymbol{\pi}} a_g(\boldsymbol{\pi}) + a_g(\boldsymbol{\pi}) + c \\
&= \mathbb{E}_q[\boldsymbol{\eta}_g(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega})]^T \nabla_{\boldsymbol{\pi}} a_g(\boldsymbol{\pi}) - \boldsymbol{\pi}^T \nabla_{\boldsymbol{\pi}} a_g(\boldsymbol{\pi}) + a_g(\boldsymbol{\pi}) + c.
\end{aligned} \tag{16}$$

where we have used the exponential family identity, $\mathbb{E}_q[t(\boldsymbol{\xi})] = \nabla_{\boldsymbol{\pi}} a_g(\boldsymbol{\pi})$. The log-normaliser term $-\mathbb{E}_q[a_g(\boldsymbol{\eta}_g(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega}))]$ is absorbed by the constant due to its independence of $q(\boldsymbol{\xi})$. In order to compute the coordinate ascent update, we need to take the gradient of $\mathcal{F}(\boldsymbol{\pi})$,

$$\nabla_{\boldsymbol{\pi}} \mathcal{F} = \nabla^2_{\boldsymbol{\pi}} a_g(\boldsymbol{\pi})(\mathbb{E}_q[\boldsymbol{\eta}_g(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega})] - \boldsymbol{\pi}). \tag{17}$$

Setting this equal to zero we find that $\boldsymbol{\pi} = \mathbb{E}_q[\boldsymbol{\eta}_g(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega})]$. Updating the parameter in this

manner, whilst keeping all other variational parameters the same, optimises the ELBO over $\boldsymbol{\pi}$.

The gradient of $\mathcal{F}(\boldsymbol{\pi})$ with respect to the local variational parameters $\sigma_{n,j}$ is given by (Hoffman et al. 2012)

$$\nabla_{\sigma_{n,j}}\mathcal{F} = \nabla^2_{\sigma_{n,j}} a_l(\sigma_{n,j})(\mathbb{E}_q[\boldsymbol{\eta}_l(y_n, v_{n,-j}, \boldsymbol{\xi})] - \sigma_{n,j}). \tag{18}$$

This leads us to find that $\sigma_{n,j} = \mathbb{E}_q[\boldsymbol{\eta}_l(y_n, v_{n,-j}, \boldsymbol{\xi})]$. The computation of the local update should be much quicker than that of the global update because whilst the global update depends on all of the local variational parameters, the local update is only dependent upon the global updates and the $n^{th}$ local parameters.

These updates form the basis of the coordinate ascent variational inference algorithm which iteratively updates all of the local paramaters and the global parameters. Algorithm 2 presents this algorithm in full and ensures that a local maximum of the ELBO is always attained (Hoffman et al. 2012).

**Algorithm 2:** Coordinate ascent mean-field variational inference

1 Randomly choose initial value for $\boldsymbol{\pi}^{(0)}$
2 **repeat**
3  **for** *each* $n \in \{1, ..., N\}, j \in \{1, ..., J\}$ **do**
4  Update the local variational parameter $\sigma_{n,j}$,

$$\sigma^{(t)}_{n,j} = \mathbb{E}_{q^{(t-1)}}[\boldsymbol{\eta}_{l,j}(y_n, v_{n,-j}, \boldsymbol{\xi})]$$

5  Update the global variational parameters,

$$\boldsymbol{\pi}^{(t)} = \mathbb{E}_{q^{(t)}}[\boldsymbol{\eta}_g(\boldsymbol{y}, \boldsymbol{v})]$$

6 **until** *the ELBO converges*

Notice that since the local update is dependent only on the global updates and the $n^{th}$ local parameters, step 5 can be done in parallel across multiple processors and the results combined

for updating the global parameters in step 6. Whilst this property of the local update can lead to more efficient computation, it also highlights an inefficiency in the algorithm. Before beginning any iterations the algorithm has to analyse every observation in the dataset using the randomly initialised global parameters. This is could be unnecessary if we were able to update the global parameters using just a subset of the dataset. This is something that stochastic variational inference exploits. It constantly improves the estimates of the global parameters as more and more observations are analysed. It does this by performing the traditional coordinate ascent updates from Algorithm 2 on a fabricated dataset consisting of a single sampled observation repeated $N$ times. This leads to intermediate global parameters which along with the previous estimate form a weighted average that becomes the new global parameters. This is much more efficient because we only perform computations on one data point at each iteration, rather than the entire dataset.

## 3.3 Natural gradient descent

Natural gradient descent is a type of optimisation method which uses the information geometry of a function's parameter space and using a Riemannian metric, it alters the direction of the conventional gradient (Hoffman et al. 2012). Typically, finding the maximum of a function $f(\boldsymbol{\pi})$ using a gradient method is done by taking steps of size $\rho$ in the direction of the gradient,

$$\boldsymbol{\pi}^{(t+1)} = \boldsymbol{\pi}^{(t)} + \rho \nabla_{\boldsymbol{\pi}} f(\boldsymbol{\pi}^{(t)}). \tag{19}$$

The gradient $\nabla_{\boldsymbol{\pi}} f(\boldsymbol{\pi})$ points in the direction of the steepest ascent, i.e. the same direction as the solution to

$$\arg\max_{d\boldsymbol{\pi}} f(\boldsymbol{\pi} + d\boldsymbol{\pi}) \quad \text{subject to } ||d\boldsymbol{\pi}||^2 < \epsilon^2 \tag{20}$$

for small enough $\epsilon$. The direction of this gradient is dependent on the Euclidean distance metric which may not be able to measure a useful notion of distance between different values of $\boldsymbol{\pi}$. The natural gradient avoids this problem by using a different definition of the gradient (Amari 1998). The symmetrised Kullback-Liebler divergence allows one to measure the dissimilarity between two probability distributions (Kullback & Leibler 1951). It is defined as,

$$D_{KL}^{\text{sym}}(\boldsymbol{\pi}, \boldsymbol{\pi}') = \mathbb{E}_{\boldsymbol{\pi}} \left[ \log \frac{q(\boldsymbol{\xi} \mid \boldsymbol{\pi})}{q(\boldsymbol{\xi} \mid \boldsymbol{\pi}')} \right] + \mathbb{E}_{\boldsymbol{\pi}'} \left[ \log \frac{q(\boldsymbol{\xi} \mid \boldsymbol{\pi}')}{q(\boldsymbol{\xi} \mid \boldsymbol{\pi})} \right]. \tag{21}$$

The symmetrised KL divergence is invariant to parameter transformations hence it is dependent only on the distributions themselves. With the symmetrised KL we can define the direction of steepest ascent as,

$$\arg\max_{d\boldsymbol{\pi}} f(\boldsymbol{\pi} + d\boldsymbol{\pi}) \quad \text{subject to } D_{KL}^{\text{sym}}(\boldsymbol{\pi}, \boldsymbol{\pi} + d\boldsymbol{\pi}) < \epsilon. \tag{22}$$

As $\epsilon$ goes to zero, the direction of the solution to equation 22 is the same as that of the *natural gradient* which takes the steepest ascent in Riemannian space. We use a Riemannian metric $I(\boldsymbol{\pi})$ to take care of the more complex constraint (do Carmo 1992). This metric is defined such that (Hoffman et al. 2012),

$$d\boldsymbol{\pi}^T I(\boldsymbol{\pi}) d\boldsymbol{\pi} = D_{KL}^{\text{sym}}(\boldsymbol{\pi}, \boldsymbol{\pi} + d\boldsymbol{\pi}). \tag{23}$$

The natural gradient can be calculated as follows (Amari 1998),

$$\hat{\nabla}_{\boldsymbol{\pi}} f(\boldsymbol{\pi}) \triangleq I(\boldsymbol{\pi})^{-1} \nabla_{\boldsymbol{\pi}} f(\boldsymbol{\pi}), \tag{24}$$

where I is the Fisher information matrix of $q(\boldsymbol{\pi})$. This is defined as (Kullback & Leibler 1951),

$$I(\boldsymbol{\pi}) = \mathbb{E}_{\boldsymbol{\pi}}\left[(\nabla_{\boldsymbol{\pi}} \log q(\boldsymbol{\xi} \mid \boldsymbol{\pi}))(\nabla_{\boldsymbol{\pi}} \log q(\boldsymbol{\xi} \mid \boldsymbol{\pi}))^T\right]. \tag{25}$$

If we restrict $q(\boldsymbol{\xi} \mid \boldsymbol{\pi})$ to be in the exponential family, then the metric is the second derivative of the log normaliser (Hoffman et al. 2012),

$$\begin{aligned} I(\boldsymbol{\pi}) &= \mathbb{E}_{\boldsymbol{\pi}}\left[(\nabla_{\boldsymbol{\pi}} \log q(\boldsymbol{\xi} \mid \boldsymbol{\pi}))(\nabla_{\boldsymbol{\pi}} \log q(\boldsymbol{\xi} \mid \boldsymbol{\pi}))^T\right] \\ &= \mathbb{E}_{\boldsymbol{\pi}}\left[(t(\boldsymbol{\xi}) - \mathbb{E}_{\boldsymbol{\pi}}\left[t(\boldsymbol{\xi})\right])(t(\boldsymbol{\xi}) - \mathbb{E}_{\boldsymbol{\pi}}\left[(t(\boldsymbol{\xi})\right])^T\right] \\ &= \nabla^2_{\boldsymbol{\pi}} a_g(\boldsymbol{\pi}). \end{aligned} \tag{26}$$

Note that we have used the property of the exponential family, that $\mathrm{Var}(t(\xi)) = \nabla^2_{\boldsymbol{\pi}} a_g(\boldsymbol{\pi})$. We shall now derive the natural gradient of the ELBO with respect to the variational parameters. We know that $\boldsymbol{\pi}$ is a natural parameter to a distribution from the exponential family hence the Fisher metric $I(\boldsymbol{\pi})$ defined by $q(\boldsymbol{\xi})$ is given by $\nabla^2_{\boldsymbol{\pi}} a_g(\boldsymbol{\pi})$. We can find the natural gradient of the ELBO with respect to the global variational parameters by pre-multiplying the gradient by the inverse Fisher information as in Equation 24 (Hoffman et al. 2012),

$$\begin{aligned} \hat{\nabla}_{\boldsymbol{\pi}} \mathcal{F} &= I(\boldsymbol{\pi})^{-1} \nabla^2_{\boldsymbol{\pi}} a_g(\boldsymbol{\pi})(\mathbb{E}_q[\boldsymbol{\eta}_g(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega})] - \boldsymbol{\pi}) \\ &= \mathbb{E}_{\boldsymbol{\sigma}}[\boldsymbol{\eta}_g(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega})] - \boldsymbol{\pi}. \end{aligned} \tag{27}$$

Similarly, the natural gradient with respect to the local variational parameters is computed as,

$$\hat{\nabla}_{\sigma_{n,j}} \mathcal{F} = \mathbb{E}_{\boldsymbol{\pi}, \boldsymbol{\sigma}_{n,-j}}[\boldsymbol{\eta}_l(\boldsymbol{y}_n, \boldsymbol{v}_{n,-j}, \boldsymbol{\xi})] - \sigma_{n,j}. \tag{28}$$

These natural gradients resemble the coordinate ascent updates we computed in Subsection 3.2. Therefore given the values of $\boldsymbol{\pi}$ and $\boldsymbol{\sigma}$, we are able to calculate the natural gradients using Equations 27 and 28.

## 3.4 Stochastic variational inference

Stochastic variational inference uses stochastic optimisation to find estimates of the global variational parameters by continually sub-sampling from the data in order to find noisy estimates of the natural gradient of the ELBO. Noisy estimates tend to be less computationally expensive than the true gradient with the added bonus that following these estimates can often lead to algorithms escaping shallow local optima of the objective function. These algorithms are proven to converge to an optimum, subject to some conditions (Robbins & Monro 1951).

Let us first introduce an objective function $f(\boldsymbol{\pi})$ along with a random function $B(\boldsymbol{\pi})$ with $\mathbb{E}_q[B(\boldsymbol{\pi})] = \nabla_{\boldsymbol{\pi}} f(\boldsymbol{\pi})$. The stochastic gradient algorithm optimises $f(\boldsymbol{\pi})$ by using repeated realisations of $B(\boldsymbol{\pi})$. For each iteration $t$, we update $\boldsymbol{\pi}$ as follows (Hoffman et al. 2012),

$$\boldsymbol{\pi}^{(t)} = \boldsymbol{\pi}^{(t-1)} + \rho_t b_t(\boldsymbol{\pi}^{(t-1)}), \tag{29}$$

where $b_t$ is a sample from $B$. Our $\boldsymbol{\pi}^{(t)}$ will converge to the optimal $\boldsymbol{\pi}^*$ as long as $f$ is convex and the following conditions on the step sizes are satisfied,

$$\sum \rho_t = \infty \text{ and } \sum \rho_t^2 < \infty. \tag{30}$$

These results also hold if we pre-multiply $b_t$ by a sequence of positive-definite matrices $I_t^{-1}$ with bounded eigenvalues (Bottou 1998):

$$\boldsymbol{\pi}^{(t)} = \boldsymbol{\pi}^{(t-1)} + \rho_t I_t^{-1} b_t(\boldsymbol{\pi}^{(t-1)}) \tag{31}$$

We can use the Fisher metric for $I_t$ which means that we will be using stochastic natural gradients instead of stochastic Euclidean gradients.

Stochastic variational inference is outlined in Algorithm 3 (Hoffman et al. 2012). We will now prove the equivalence between this algorithm and stochastic natural gradient ascent on the global variational parameters. We shall do this by finding global variational parameters such

**Algorithm 3:** STOCHASTIC VARIATIONAL INFERENCE

1 Randomly choose initial value for $\boldsymbol{\pi}^{(0)}$
2 Choose a step-size sequence $\rho_t$
3 **repeat**
4 Take a sample of an observation $y_i$ from the dataset.
5 Update the local variational parameter,

$$\sigma = \mathbb{E}_{\boldsymbol{\pi}^{(t-1)}}[\boldsymbol{\eta}_g(y_i^{(N)}, \boldsymbol{v}_i^{(N)})].$$

6 Update the intermediate global parameters using a vector of $y_i$ repeated $N$ times,

$$\hat{\boldsymbol{\pi}} = \mathbb{E}_{\sigma}[\boldsymbol{\eta}_g(y_i^{(N)}, \boldsymbol{v}_i^{(N)})].$$

7 Compute new estimate of the global variational parameters,

$$\boldsymbol{\pi}^{(t)} = (1 - \rho_t)\boldsymbol{\pi}^{(t-1)} + \rho_t \hat{\boldsymbol{\pi}}$$

8 **until** *forever*

that the ELBO is maximised. We can write the ELBO $\mathcal{F}(\boldsymbol{\pi}, \sigma)$ in terms of both the local and global variational parameters. Let $\sigma(\boldsymbol{\pi})$ be a local optimum of the local parameters, i.e. $\nabla_\sigma \mathcal{F}(\boldsymbol{\pi}, \sigma(\boldsymbol{\pi})) = 0$. Let us define the *locally maximised ELBO* $\mathcal{F}(\boldsymbol{\pi}) \triangleq \mathcal{F}(\boldsymbol{\pi}, \sigma(\boldsymbol{\pi}))$ as the ELBO with fixed $\boldsymbol{\pi}$ and local parameters set to the local optimum $\sigma(\boldsymbol{\pi})$. In order to find the natural gradient of $\mathcal{F}(\boldsymbol{\pi})$ we can make use of the following property:

$$\begin{aligned}\nabla_{\boldsymbol{\pi}} \mathcal{F}(\boldsymbol{\pi}) &= \nabla_{\boldsymbol{\pi}} \mathcal{F}(\boldsymbol{\pi}, \sigma(\boldsymbol{\pi})) + (\nabla_{\boldsymbol{\pi}} \sigma(\boldsymbol{\pi}))^T \nabla_\sigma \mathcal{F}(\boldsymbol{\pi}, \sigma(\boldsymbol{\pi})) \\ &= \nabla_{\boldsymbol{\pi}} \mathcal{F}(\boldsymbol{\pi}, \sigma(\boldsymbol{\pi}))\end{aligned} \tag{32}$$

where $\nabla_{\boldsymbol{\pi}}\sigma(\boldsymbol{\pi})$ is the Jacobian of $\sigma(\boldsymbol{\pi})$. The ELBO $\mathcal{F}(\boldsymbol{\pi})$ can be decomposed as follows (Hoffman et al. 2012),

$$\mathcal{F}(\boldsymbol{\pi}) = \mathbb{E}_q[\log p(\boldsymbol{\xi})] - \mathbb{E}_q[\log q(\boldsymbol{\xi})] + \sum_{n=1}^{N} \max_{\sigma_n}(\mathbb{E}_q[\log p(y_n, \boldsymbol{v}_n \mid \boldsymbol{\xi})] - \mathbb{E}_q[\log q(\boldsymbol{v}_n)]). \quad (33)$$

We need to incorporate the randomly chosen samples from the dataset that are used to compute noisy estimates of the natural gradient. To do this, we let $I$ be a variable that randomly chooses an index between 1 and $N$, $I \sim \text{Unif}(1, ..., N)$. Next we define

$$\mathcal{F}_I(\boldsymbol{\pi}) \triangleq \mathbb{E}_q[\log p(\boldsymbol{\xi})] - \mathbb{E}_q[\log q(\boldsymbol{\xi})] + N \max_{\sigma_I}(\mathbb{E}_q[\log p(y_I, \boldsymbol{v}_I \mid \boldsymbol{\xi})] - \mathbb{E}_q[\log q(\boldsymbol{v}_I)]). \quad (34)$$

Clearly we have that $\mathbb{E}[\mathcal{F}_I(\boldsymbol{\pi})] = \mathcal{F}(\boldsymbol{\pi})$ and so $\hat{\nabla}_{\boldsymbol{\pi}}\mathcal{F}_I(\boldsymbol{\pi})$ is a noisy but unbiased estimate of $\hat{\nabla}_{\boldsymbol{\pi}}\mathcal{F}(\boldsymbol{\pi})$. Sampling a single observation with index $i$ from the dataset and repeating it N times, it follows that $\mathcal{F}_i(\boldsymbol{\pi}) = \mathcal{F}(\boldsymbol{\pi})$. Therefore, $\hat{\nabla}\mathcal{F}_i(\boldsymbol{\pi})$ can be computed using Equation 27,

$$\hat{\nabla}\mathcal{F}_i = \mathbb{E}_q\left[\boldsymbol{\eta}_g\left(y_i^{(N)}, \boldsymbol{v}_i^{(N)}, \boldsymbol{\omega}\right)\right] - \boldsymbol{\pi} \quad (35)$$

where $\{y_i^{(N)}, \boldsymbol{v}_i^{(N)}\}$ is the dataset created by replicating observation $y_i$ and latent variable $\boldsymbol{v}_i$, $N$ times. Equation 35 can be written in more detail. Using the expression for $\boldsymbol{\eta}_g(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\omega})$ in Equation 7, we can find the conditional natural parameter for the global variational parameter using our sample replicated $N$ times (Hoffman et al. 2012),

$$\boldsymbol{\eta}_g\left(y_i^{(N)}, \boldsymbol{v}_i^{(N)}, \boldsymbol{\omega}\right) = \boldsymbol{\omega} + N.(t(y_n, \boldsymbol{v}_n), 1). \quad (36)$$

Substituting for this into Equation 27, we compute the noisy natural gradient as

$$\hat{\nabla}_{\boldsymbol{\pi}}\mathcal{F}_i = \boldsymbol{\omega} + N.(\mathbb{E}_{\sigma_i(\boldsymbol{\pi})}[t(y_i, \boldsymbol{v}_i)], 1) - \boldsymbol{\pi} \quad (37)$$

where $\sigma_i(\boldsymbol{\pi})$ denotes the components of $\sigma(\boldsymbol{\pi})$ that pertain to the $i^{\text{th}}$ observation. This is computationally much cheaper since it only uses the local parameters of one data point compared to those of the entire dataset for the full natural gradient.

Lastly, we implement a Robbins-Monro algorithm which optimises the ELBO (Robbins & Monro 1951). At each iteration, we take a sample observation, compute the intermediate global parameter as the estimate of $\boldsymbol{\pi}$ that we would get if we replicated the sample $N$ times (Hoffman et al. 2012):

$$\hat{\boldsymbol{\pi}}_t \triangleq \boldsymbol{\omega} + N\mathbb{E}_{\sigma_i(\boldsymbol{\pi})}[t(y_i, \boldsymbol{v}_i), 1]. \tag{38}$$

Having computed the intermediate global parameter, we use the noisy gradient to update the global parameter:

$$\boldsymbol{\pi}^{(t)} = \boldsymbol{\pi}^{(t-1)} + \rho_t(\hat{\boldsymbol{\pi}}_t - \boldsymbol{\pi}^{(t-1)}) \tag{39}$$

$$= (1 - \rho_t)\boldsymbol{\pi}^{(t-1)} + \rho_t\hat{\boldsymbol{\pi}}_t. \tag{40}$$

Clearly, this updates the global parameter as a weighted average of the current parameter and the recently calculated intermediate global parameter. We define the step-size at iteration $t$ as $\rho_t = (t + \tau)^{-\kappa}$. The *forgetting rate* $\kappa \in (0.5, 1]$ dictates the speed at which the algorithm forgets the old information. The *delay* $\tau \geq 0$ decreases the weight of the earlier iterations.

# 4 Stochastic variational inference for LDA

We shall now discuss the use of stochastic variational inference for the purpose of estimating the posterior distribution of the latent variables conditioned on the observations in latent Dirichlet allocation. We introduced this model in Section 2, where we discussed how the model assumes that each document is a distribution over a set of topics and the topics

themselves are distributions over the words in the corpus.

It is important to outline the global and local variables involved so that the model is in the same framework as was set up in Section 3. The global latent variables are the topics $\boldsymbol{\beta}_k$. The local latent variables are the topic proportions $\boldsymbol{\theta}_d$ and the topic assignments $z_{d,n}$. Lastly the observations are the words in each document $w_{d,n}$ (Hoffman et al. 2012). The local variables are placed in the local context of a document $d$. The plate notation for LDA with stochastic variational inference is illustrated in Figure 2. The variables inside squares are the variational parameters for each of the latent variables.

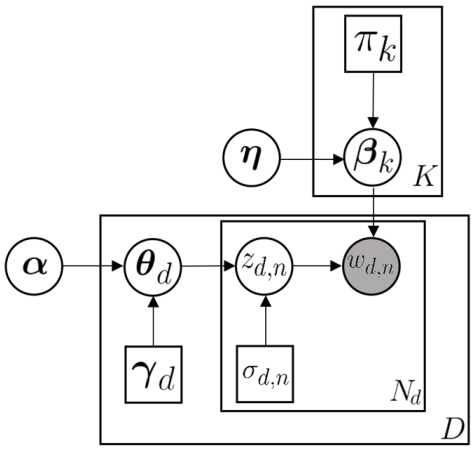


Figure 2: Plate notation for Latent Dirichlet allocation with stochastic variational inference.

Since we assumed that our variational distribution is in the mean-field family, the variational distribution of the latent variables has the form:

$$\prod_{k=1}^{K} q(\boldsymbol{\beta}_k; \pi_k) \prod_{d=1}^{D} \left\{ q(\boldsymbol{\theta}_d; \gamma_d) \prod_{n=1}^{N_d} q(z_{d,n}; \sigma_{d,n}) \right\} \tag{41}$$

In Section 3 we introduced the complete conditional distribution which is the distribution of a variable conditioned on all of the other variables in the model. We also discussed mean-field variational inference in which the variational distribution and complete conditional for each latent variable are assumed to be members of the same family. The complete conditional of

the topic assignment is given by (Hoffman et al. 2012)

$$p(z_{d,n} = k \mid \boldsymbol{\theta}_d, \boldsymbol{\beta}, w_{d,n}) \propto \theta_{d,k}\beta_{k,w_{d,n}} = \exp\{\log \theta_{d,k} + \log \beta_{k,w_{d,n}}\}. \tag{42}$$

This is a multinomial distribution hence it has a multinomial variational distribution, $q(z_{d,n}) = \text{Multinomial}(\sigma_{d,n})$. Each word is given a different variational distribution for its topic allocation. The complete conditional of the topic proportions is given by

$$p(\boldsymbol{\theta}_d \mid \boldsymbol{z}_d) = \text{Dirichlet}(\boldsymbol{\alpha} + \sum_{n=1}^{N_d} z_{d,n}). \tag{43}$$

Clearly this is a Dirichlet distribution although it is no longer an exchangeable Dirichlet. Since this conditional is Dirichlet, its variational distribution is also Dirichlet, $q(\boldsymbol{\theta}_d) = \text{Dirichlet}(\gamma_d)$. Each document has its own Dirichlet parameter. This allows every document to contain different topics in different proportions. The complete conditional for the topics is given by

$$p(\boldsymbol{\beta}_k \mid \boldsymbol{z}, \boldsymbol{w}) = \text{Dirichlet}(\boldsymbol{\eta} + \sum_{d=1}^{D}\sum_{n=1}^{N_d} z_{d,n}^k w_{d,n}) \tag{44}$$

Again, since the complete conditional is Dirichlet, the variational distribution is also Dirichlet, $q(\boldsymbol{\beta}_k) = \text{Dirichlet}(\pi_k)$.

We can now use these complete conditional distributions to find the coordinate ascent updates. We do this by computing the expectation of the natural parameter of each complete conditional. Coordinate ascent variational inference iterates in two steps. First it updates both the local parameters $\{\sigma_{d,n}, \gamma_d\}$ and secondly it updates the global parameters $\pi_k$. The

local variational parameters are updated as follows,

$$\begin{aligned}\sigma_{d,n}^{k} &= \exp\{\mathbb{E}_{\gamma_d}[\log\theta_{d,k}] + \mathbb{E}_{\pi_k}[log\beta_{k,w_{d,n}}]\} \\ &\propto \exp\{\Psi(\gamma_{d,k}) + \Psi(\pi_{k,w_{d,n}}) - \Psi(\sum_{v}\pi_{k,v})\} \quad \text{for } n \in \{1,...,N_d\} \end{aligned} \tag{45}$$

$$\boldsymbol{\gamma}_d = \mathbb{E}[\boldsymbol{\alpha} + \sum_{n=1}^{N_d} z_{d,n}] = \boldsymbol{\alpha} + \sum_{n=1}^{N_d}\sigma_{d,n}, \tag{46}$$

where $\Psi$ is the first derivative of the log Gamma function. Equation 45 uses the Dirichlet expectation: $\mathbb{E}[\log\boldsymbol{\theta}_k \mid \boldsymbol{\gamma}] = \Psi(\gamma_k) - \Psi(\sum_{i=1}^{K}\gamma_i)$. Equation 46 uses the property of an indicator, $\mathbb{E}_q[z_{d,n}^k] = \sigma_{d,n}^k$.

Having updated the local parameters, i.e. those within each document, we update the global parameter. This is the variational Dirichlet for the topics:

$$\pi_k = \mathbb{E}\left[\boldsymbol{\eta} + \sum_{d=1}^{D}\sum_{n=1}^{N_d} z_{d,n}^k w_{d,n}\right] = \boldsymbol{\eta} + \sum_{d=1}^{D}\sum_{n=1}^{N_d}\sigma_{d,n}^k w_{d,n}. \tag{47}$$

As we can see the update is dependent on the variational parameters from all of the documents. This can make performing the update very computationally costly. Stochastic variational inference makes updating the global variational parameters much more efficient by sub-sampling just one document from the dataset per iteration, calculating its local variational parameters and then computing intermediate global variational parameters (topics) by forming a collection of the sample repeated $N = D$ times. These topics are computed using the following:

$$\hat{\pi}_k = \boldsymbol{\eta} + D\sum_{n=1}^{N_d}\sigma_{d,n}^k w_{d,n}. \tag{48}$$

The updated topics are then a weighted average of the intermediate and current topics:

$$\pi_k^{(t+1)} = (1-\rho_t)\pi_k^{(t)} + \rho_t \hat{\pi}_k. \tag{49}$$

Stochastic variational inference for the LDA model is described in Algorithm 4 (Hoffman et al. 2012)

**Algorithm 4:** Stochastic variational inference for LDA

Randomly choose initial value for $\boldsymbol{\pi}^{(0)}$
Choose a step-size sequence $\rho_t$
**repeat**
  Take a sample document $w_d$ from the dataset.
  Initialise $\gamma_{d,k} = 1$ for $k \in \{1, ..., K\}$.
  **repeat**
    **for** $n \in \{1, ..., N_d\}$ **do**
      Set

$$\sigma_{d,n}^k \propto \exp\{\Psi(\gamma_{d,k}) + \Psi(\pi_{k,w_{d,n}}) - \Psi(\textstyle\sum_v \pi_{k,v})\} \quad \text{for } k \in \{1, ..., K\}$$

    Set $\boldsymbol{\gamma}_d = \boldsymbol{\alpha} + \sum_{n=1}^{N_d} \sigma_{d,n}$
  **until** *local parameters* $\sigma_{d,n}$, $\gamma_d$ *converge*
  **for** $k \in \{1, ..., K\}$ **do**
    Compute the intermediate topics using a collection of $w_d$ repeated $D$ times,

$$\hat{\pi}_k = \boldsymbol{\eta} + D \sum_{n=1}^{N_d} \sigma_{d,n}^k w_{d,n}.$$

  Compute new update for the global variational parameters,

$$\boldsymbol{\pi}^{(t)} = (1-\rho_t)\boldsymbol{\pi}^{(t-1)} + \rho_t \hat{\boldsymbol{\pi}}$$

**until** *forever*

# 5 Application to BBC news article dataset

## 5.1 Background

Now that we have introduced the latent Dirichlet allocation and the stochastic variational inference algorithm used to for its posterior inference, it makes sense to apply this model to a dataset of our own. The dataset we will be using is one generated by scraping the text from a number of BBC news articles `https://www.bbc.com/news`. We have scraped the text from $D = 8422$ documents from nine different sections of the BBC news website. These are: *coronavirus, UK, world, business, politics, technology, science, health* and *family and education*. Theoretically this would suggest that the documents should be clustered into $K = 9$ topics. However, we will perform tests to decide on the optimal number of topics as we know documents can exhibit multiple topics, and these may not all be captured by the nine categories above.

## 5.2 Implementation

In order to scrape the text from an article, we need its URL. An efficient tool to use in order to access the URLs of our news articles is Web Scraper (https://webscraper.io/) which can be accessed through a google chrome extension. This enables users to scrape data from dynamic web pages, such as those with multiple levels of navigation. These include categories and pagination. We make use of their pagination handlers in order to scrape the URLs of the articles on the first 50 pages of each section of the BBC news website.

Now that we have the URLs of the 8422 articles, we need to scrape the text data from them. We shall be using *Python 3*, in particular the libraries *os*, *urllib* and*BeautifulSoup*. These three libraries can be used in tandem with urllib sending the URL requests, BeautifulSoup extracting the text and os writing this text to a new file.

With the article text data extracted, we need to perform a few pre-processing steps to make

sure that the LDA model can perform properly. These steps are common in most natural language processing tasks. We begin by replacing all of the punctuation with white space, removing all numbers and converting all words to lower case. Next we remove all stop words from the text. Stop words are common words that are considered unimportant for the overall purpose of our model. There is no standardised list of stop words but we will be using the list from the Python library *nltk*. This contains words such as *and, if, but, the* and *for*. Clearly such words provide no evidence as to the topic of a document and so they can be removed. With the documents pre-processed, the total number of words in the corpus is $N = 2395739$.

For the implementation of the latent Dirichlet allocation model with stochastic variational inference, we will use the Python library *gensim*. This is a library specifically created for topic modelling and contains functionality for models such as *word2vec* and *doc2vec*.

## 5.3 Results

There are a number of methods which we can use to analyse our fitted model. We will be looking at both intrinsic and visual methods. We introduced a number of hyperparameters in the previous sections. Therefore we should look at how the model performs for a range of different values of these hyperparameters. The hyperparameters we shall look at are; the number of topics $K$, the forgetting rate $\kappa$ and the delay $\tau$.

In order to decide on the optimal settings for these hyperparameters we need to introduce a metric with which we can evaluate each model. One such metric is the *perplexity*. This is defined as a measurement of how well a probability model predicts a sample. It is monotonically decreasing in the likelihood of the test data and so a model with better generalisation capability will score a lower perplexity. We formally define perplexity for a test set of $M$

documents as,

$$perplexity(D_{\text{test}}) = \exp\left\{-\frac{\sum_{d=1}^{D}\log p(\boldsymbol{w}_d)}{\sum_{d=1}^{D} N_d}\right\} \tag{50}$$

where, as we defined in Subsection 2.2, $\boldsymbol{w}_d$ refers to the words in document $d$. Perplexity is equivalent to the inverse of the geometric per-word likelihood thus as the number of topics $K$ increases, the perplexity should decrease. We have randomly held out 10% of the documents from our dataset to act as a test set and trained an LDA model on the remaining 90% for a range of numbers of topics from $K = 1$ to 25. The results are displayed in Figure 3.

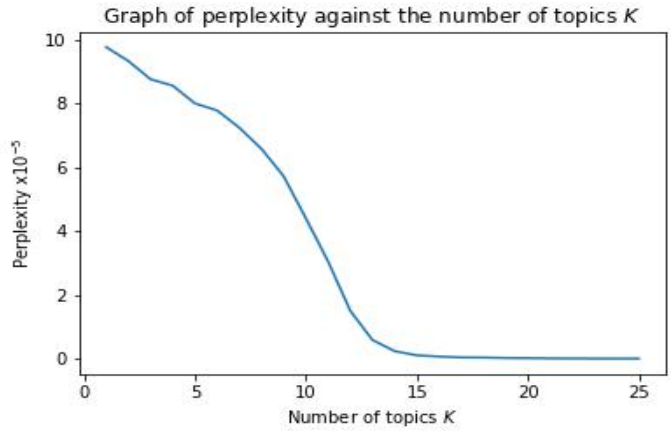


Figure 3: Graph of perplexity for different numbers of topics $K$. Elbow point at $K = 13$.

Looking at this figure we can see that, as expected, the perplexity decreases as the number of topics increases. The elbow point (the point at which the increase in optimization function is no longer worth the increased cost) of the model appears to be at $K = 13$ which is surprising given that we have collected articles from 9 different sections of the BBC news website. However, we will go forward with the $K = 13$ model and move onto tuning the forgetting rate $\kappa \in (0.5, 1]$ and the delay $\tau \geq 0$.

We shall create a grid of values for these two hyperparameters and measure the perplexity of the model at each pair. We shall measure the perplexity for the following values of $\kappa$:

$\{0.6, 0.7, 0.8, 0.9, 1.0\}$. Similarly, we shall measure the perplexity for the following values of $\tau$: $\{1.0, 2.0, 3.0, 4.0\}$. Having run the model and computed the perplexity for each setting of $\kappa$ and $\tau$ we achieve the results shown in Table 2.

| log perplexity | | | | |
|---|---|---|---|---|
| | $\tau = 1$ | $\tau = 2$ | $\tau = 3$ | $\tau = 4$ |
| $\kappa = 0.6$ | -12.15 | -9.82 | -9.80 | -9.78 |
| $\kappa = 0.7$ | -12.58 | -9.82 | -9.85 | -9.80 |
| $\kappa = 0.8$ | -13.19 | -10.28 | -10.43 | -10.41 |
| $\kappa = 0.9$ | -13.64 | -11.13 | -11.33 | -11.41 |
| $\kappa = 1.0$ | -14.09 | -11.75 | -11.95 | -12.02 |

Table 2: log perplexities for tuning the hyperparameters

Looking at Table 2, the optimal tuning for the hyperparameters is $\kappa = 1.0$ and $\tau = 1$. Therefore we shall use these values along with $K = 13$ in the model going forwards.

So far we have used an intrinsic evaluation method to determine how well our models fit the dataset. Now we have settled on our hyperparameters we can use some visual methods to evaluate our model. We shall begin by looking at a summary of our dataset which is illustrated in Figure 4. The tall, thin peak on the far left indicates that a large number

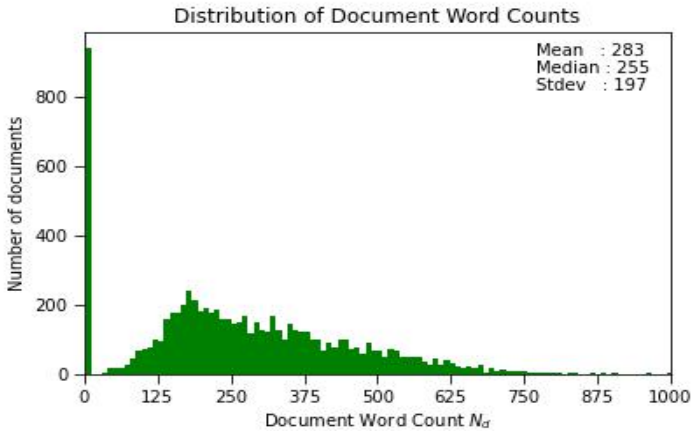


Figure 4: Histogram displaying word count against Document number

of the documents contain very few words. However there also look to be many documents

with at least $N_d = 125$ words with several small peaks on the far right suggesting that some articles have close to $N_d = 1000$ words. It will be interesting to see the difference in topic proportions for documents with a large difference in word counts. Perhaps some topics will largely be dominant in documents that have a lower word count.

With our LDA model fitted, we can look at some visualisations of the topics generated. Figure 5 illustrates the word counts of documents in their dominant topics. There appears

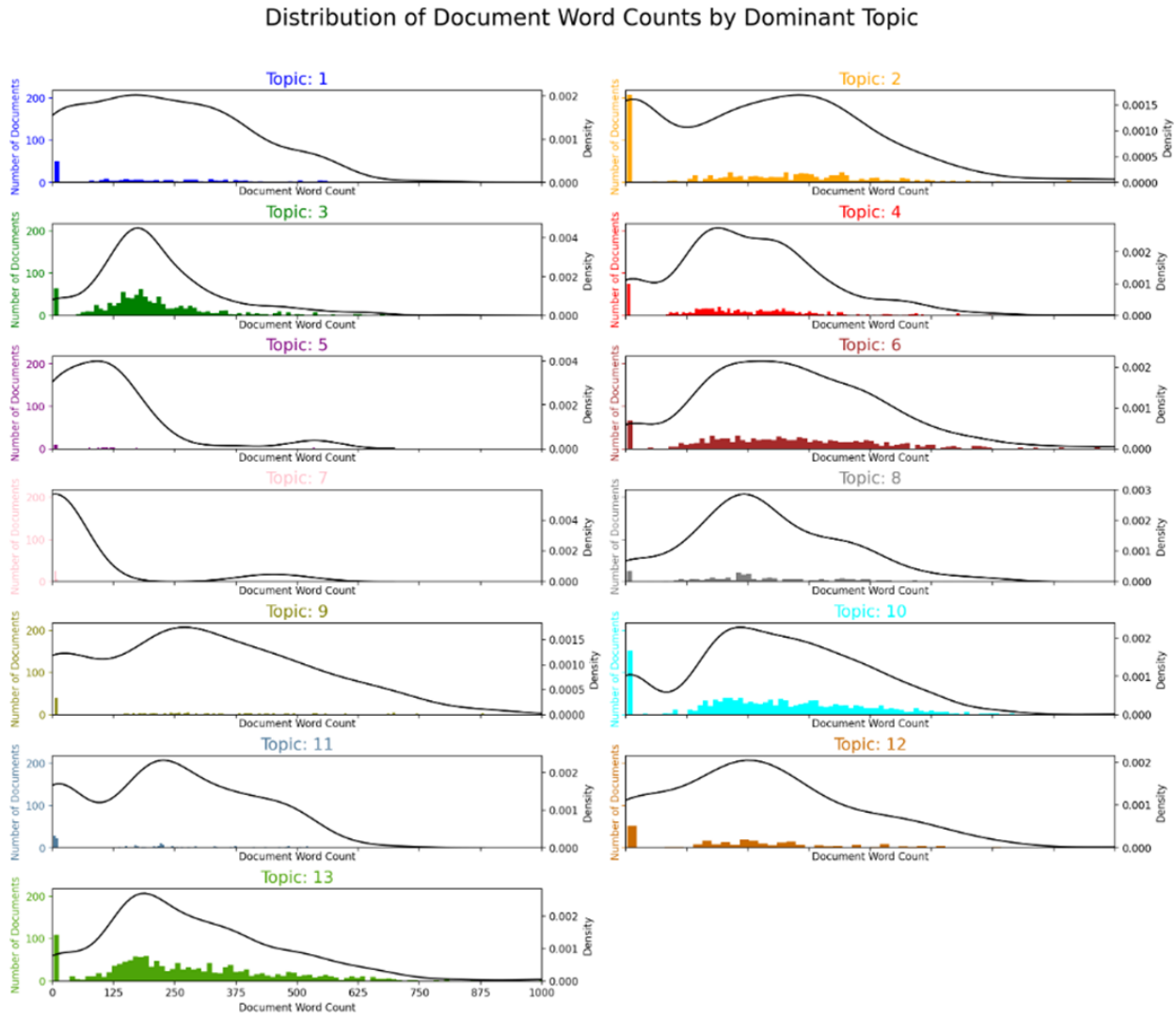


Figure 5: The distribution of the word counts within each topic where each document is shown in its dominant topic.

to be a spread in terms of document word count across most topics. As one would expect,

the initial peak appears in most topics which correlates with the peak in the same position of Figure 4. Several topics stand out in terms of the total number of documents that they are dominant in. Topics 13, 10, 6 and 3 all seem to be dominant in a large number of documents. On the contrary, topics 5, 7, 9 and 11 seem to be dominant in very few. As the right hand upper axis displays, we have also produced a representation of the topic word counts as a continuous probability density curve. The shape of this density is similar for most topics with an initial dip before a peak around $N_d = 300$. The exceptions to this are topics 1, 5 and 7 which is potentially a result of them being dominant in very few documents.

We should also take some other visualisations of our generated topics. Figure 6 displays some of the keywords for each topic. These are words which have high weight and occur frequently within each topic. Upon first inspection of Figure 6, it is clear that *Covid* is very common among our topics. Indeed, *Covid* or similar words such as *pandemic* appear in all topics. This makes sense as the pandemic is still very much at the forefront of the news with its effects still being felt across the world. It is a similar case for *health* which appears in over half of the topics. Again this makes sense since the covid pandemic has caused huge problems to both the health of people and the healthcare systems that serve them. It is no surprise to see that *UK* is a keyword in 6 topics given that the BBC is the national broadcaster of the United Kingdom and so its news articles mainly report on events in the UK. Certain topics can be assumed to represent some of the sections of the BBC news website from which our dataset was taken. The *World* section seems to appear quite heavily in topics 9 and 10 with the former containing the keywords *North, Korea* and *China* and the latter containing the keywords *Ukraine, Russia, Shanghai* and *China* as well as the word *world* itself. The *Business* section seems to be covered by topic 11 which features the keywords *work, pay* and *production*. Lastly, the *Politics* section also appears to feature in topic 2 as it contains the keywords *Boris, Johnson, prime, minister* and *MPs*. The *technology, science* and *family and education* sections do not appear to have been captured by our topics. Perhaps this would

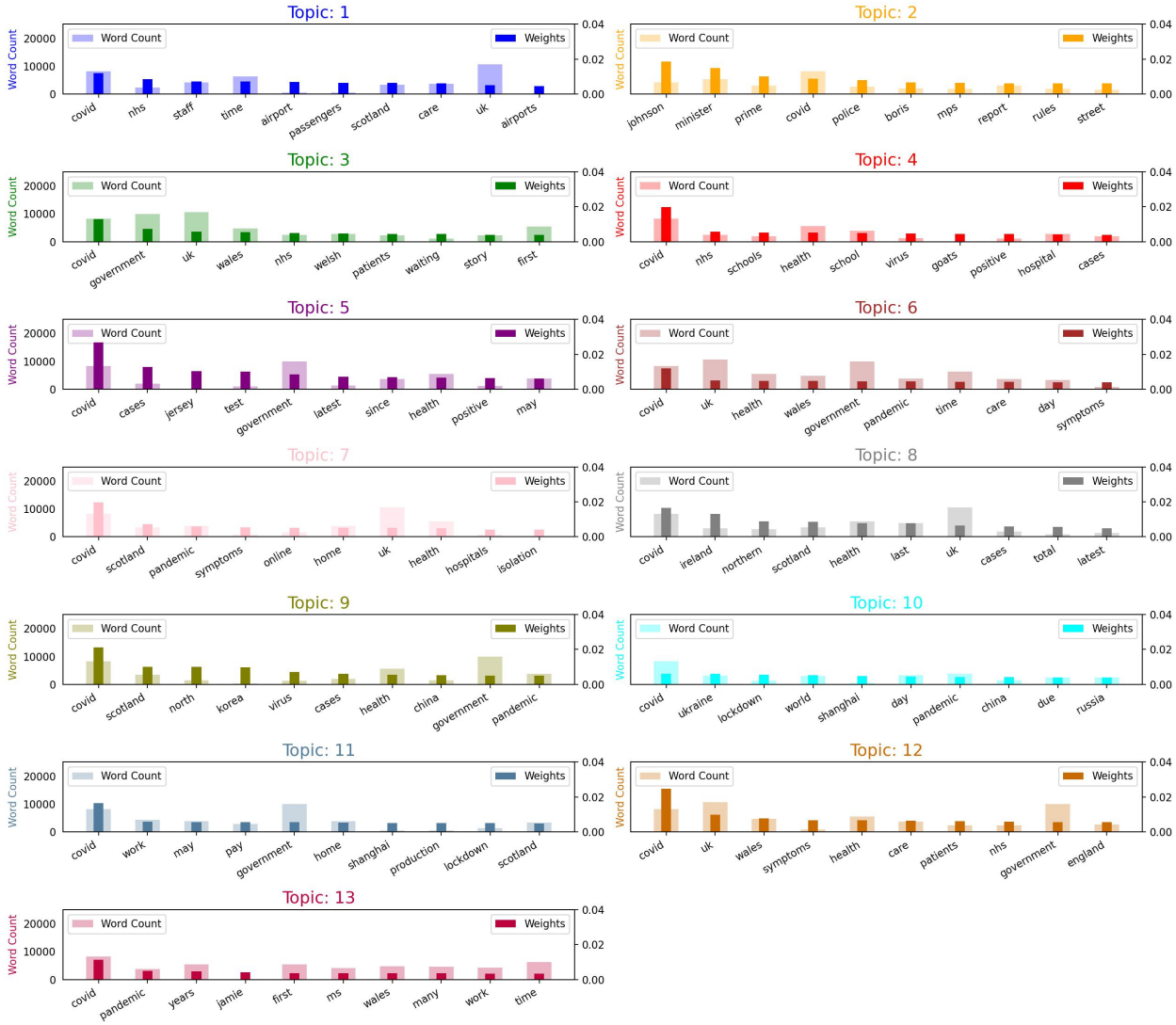


Figure 6: The word count and weights of the top 10 keywords in the 13 topics.

not be the case were we to increase the number of topics $K$. However, this would come at the expense of a greater to computational cost to running the model and so we would have to restrict the number of additional topics if possible.

It would be useful to have a look at the distribution of the documents' dominant topics and topic weightages. These are shown in Figure 7. For the graph on the left we have taken the topic with the highest weight from each document. For the graph on the right we have summed the each topic's weight over all documents. The figure indicates that several topics

are relatively equally dominant across the corpus. Topics 13 (Covid) and 10 (World) are the most common dominant topics. This is not unexpected given the aforementioned pandemic and the ongoing war in Ukraine. Certain topics are dominant in very few documents. Topics

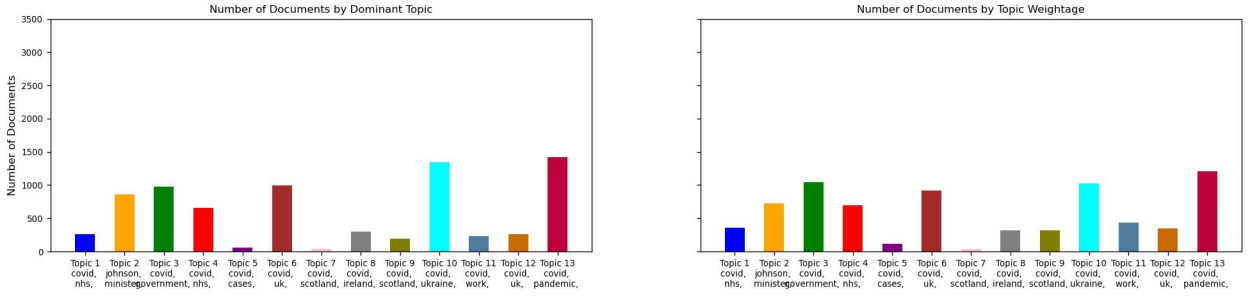


Figure 7: The distributions of the dominant topics and topic weightages

5 and 7 seem to be dominant in fewer than 100 documents. This correlates with their topic weightage in the graph on the right meaning they are missing altogether from most of the corpus. It is not surprising that both graphs look similar. This suggests that a topic will tend to either have the highest weight in a document or have a very low weight. There are potentially a few exceptions to this with topics 9 and 11 having a noticeably higher topic weightage than dominant topic count. This implies that they tend to have a relatively high weight even if they are not the dominant topic in a document.

A useful method for visualising high-dimensional data is t-distributed stochastic neighbour embedding (t-SNE) (van der Maaten & Hinton 2008) which is based on stochastic neighbour embedding (Rowels & Hinton 2002). It is a non-linear dimensionality reduction technique which is particularly useful for representing data in 2 or 3 dimensions allowing for easier visualisation. t-SNE works by constructing a probability distribution over every pair of points in the high dimension with higher probability going to similar points. It then attempts to create a similar probability distribution in the lower dimension, by minimising the Kullback-Liebler divergence between the two distributions. Figure 8 contains a t-SNE visualisation of our LDA topics in 2 dimensions. The colour of each point represents the dominant topic within each document and the specific colour representing each topic is the same as in the

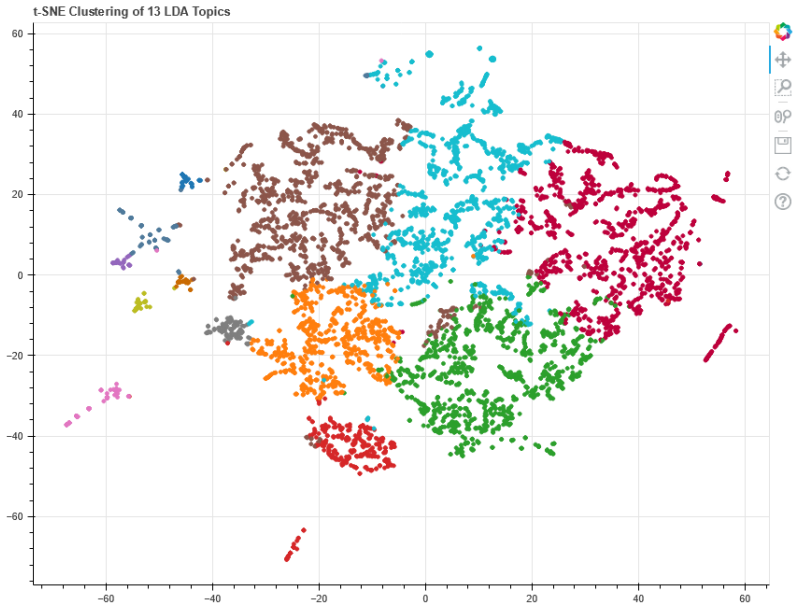


Figure 8: t-SNE visualisation of our LDA model

previous figures. Upon first inspection we can see some degree of separation between clusters with some clusters positioned quite some distance away from the main bulk of data points. The clusters themselves do not look extremely dense but this is not too important given that the t-SNE clusters are not proportionally related to the clusters in the high-dimensional space. More important is the distance and location of each cluster in relation to one another. Those clusters that are close to each other will be more closely related to one another. So looking at the figure we can see which topics are related. For example topics 13 (rose red) and 3 (green) are close to one another. They are both topics containing keywords around the subject of covid and the UK so this makes sense. On the other hand, we can also determine which topics are dissimilar by looking at topics that are further apart on the t-SNE plot. For example, topics 10 (turquoise) and 7 (pink) are quite far apart and so they are clearly not dominant in many of the same documents. This makes sense given that topic 10 mainly contains keywords surrounding the subject of world politics whereas topic 7's keywords are mainly relate to the UK and the effect the pandemic is having on it.

# 6 Conclusion

In this report we have discussed one of the most commonly used topic models in the field of natural language processing - the latent Dirichlet allocation model. This generative statistical model generates a collection of topics within a corpus which are distributed over the observed words in the corpus. The individual documents are generated as a distribution over these topics with each word in each document being allocated to a topic. In order to generate the topics, LDA requires the posterior distribution $p(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z} \mid \boldsymbol{w})$ which is the distribution of the topics, topic proportions and topic allocations conditioned on the observed words. However this distribution is intractable to compute and so we have to find a suitable approximation. One approach to approximating the posterior is stochastic variational inference. This combines stochastic optimisation with variational inference. Variational inference is used to approximate a distribution by proposing a family of distributions over the latent variables and choosing the member of the family that is closest to the posterior. Specifically, we use mean-field variational inference in which we make the assumption that the variational distribution can be factorised over a partition of the latent variables. The variational inference algorithm we use makes use of coordinate ascent which iteratively updates the local and global variational parameters - the parameters which our latent variables are dependent on. We used stochastic optimisation to optimise the variational objective function using noisy estimates of its natural gradient. These estimates are noisy due to the fact the algorithm only takes a sample of one observation from the dataset rather than using the entire dataset. Having discussed the methods involved in performing LDA, we moved onto our application. We generated a dataset by extracting the URLs of 8422 articles on the BBC news website. With the URLs in hand we could perform http requests to access the articles themselves, extracting the raw html content and saving this into individual files. We could then go through these html files and extract only the text elements. We then pre-processed the data, removing all punctuation and numbers, taking out any commonly used stop words and

saved all of the data in one large array. With the dataset ready we could begin running LDA models on it. We used the python library Gensim to implement this. In order to get the best results, we needed to decide on the optimal number of topics for the model. Thus for a range of values of $K \in \{1, ..., 25\}$, we trained an LDA model on 90% of the dataset and held out the other 10% to test it on. Upon looking at a plot of the perplexity of the model for the different values of $K$, the optimum appeared to be at $K = 13$. Therefore we could move onto tuning the two hyperparameters of the model; the forgetting rate (dictates the speed at which the algorithm forgets the old information) and the delay (decreases the weight of the earlier iterations). To do this, we created a grid of values for the forgetting rate (0.6, 0.7, 0.8, 0.9, 1.0) and decay (1.0, 2.0, 3.0, 4.0) and measured the perplexity of the model when trained and tested on the same samples of the dataset as we did when finding the optimal number of topics. This led us to find the optimal setting of (1.0,1.0). With the parameters of our model decided, we were able to look at some visualisations of the results. This began by inspecting the distribution of the document word counts which revealed that a large proportion of documents contained less than 50 words. However some documents had more than 600 words and so the average word count was 283. We then looked at the distributions of the word counts within their dominant topics which backed up what we saw with the previous figure, and also gave us insight into which topics were the most and least dominant across the corpus. Next we looked at the word counts and weights of the top 10 keywords within each topic. These revealed a common theme among the majority of the topics, with *covid* or covid-related words being present in pretty much every topic. Similarly words such as *health* and *nhs* were very common. This indicated that there was not a great deal of distinction between topics. The graph displaying the number of documents that each topic was dominant in made it clear that some topics were much more dominant than others across the corpus. Topics 3,6,10 and 13 in particular, had both high numbers of documents that they were dominant in and high weightages across the corpus. On the contrary, topics

5 and 7 had very low dominance.

Lastly we looked at a t-SNE visualisation of our topic proportions across the documents. This took our 13-dimensional results data and reduced it to just 2 dimensions for ease of visualisation. The results were interesting with many topics with similar themes being close to each other in the lower dimension.

So what conclusions can we draw about our dataset having fit our LDA model to it? We can see that there are a number of different topics exhibited across the documents including covid, health, events in the UK such as UK politics and events in the world outside the UK such as those going on in Ukraine with the Russian invasion. We can clearly see that the topics we have picked up are very different from the categories on the BBC news website as many of our topics overlap in terms of those categories.

Latent Dirichlet allocation is just one of a number of topic models that are commonly used in natural language processing. Hierarchical latent tree analysis (HLTA) is another model that is commonly used to detect topics within a collection of documents (Liu et al. 2014). HLTA avoids a particular issue with LDA which is that different topics can give high weight to the same words meaning that certain words can be keywords in every topic. This was the case for us with words such as *covid* appearing in with high proportion in pretty much every topic. HLTA avoids this issue by modelling topics using words that have high frequency in that topic and low frequency in all other topics. Perhaps in the future, this project could be extended by fitting a HLTA model to our BBC news dataset in order to try and establish some more distinct topics. Additionally, one could run a completely different type of model on the dataset. Perhaps a semantic analysis model such as VADER (Hutto & Gilbert 2014). Semantic analysis models attempt to determine the subjective information conveyed through text data. This would allow one to determine the polarity of each article in the dataset and whether it is a positive or negative piece of news. Of course, our dataset is only a small subset of the articles available on the BBC news website. In reality there are hundreds of

thousands of articles going back decades which could be modelled by LDA. This would allow us to identify a lot more topics than those we identified, as articles on events such as the covid pandemic only go back a few years. It would be interesting to see what other topics get picked up from the last few decades. Perhaps the great recession of 2007 or the death of Princess Dianna in 1997.

# References


Alexander, D. H., Novembre, J. & Lange, K. (2009), 'Fast model-based estimation of ancestry in unrelated individuals', *Genome research* **19**(9), 1655–1664.

Amari, S.-i. (1998), 'Natural gradient works efficiently in learning', *Neural Computation* **10**(2), 251–276.
**URL:** *https://doi.org/10.1162/089976698300017746*

Bishop, C. M. (2006), *Pattern Recognition and Machine Learning*, Springer.

Blei, D. M. (2012), 'Probabilistic topic models', *Commun. ACM* **55**(4), 77â84.
**URL:** *https://doi.org/10.1145/2133806.2133826*

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

Bottou, L. (1998), On-line learning and stochastic approximations, *in* 'In On-line Learning in Neural Networks', Cambridge University Press, pp. 9–42.

Cao, L. & Fei-Fei, L. (2007), Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes, *in* '2007 IEEE 11th International Conference on Computer Vision', pp. 1–8.

do Carmo, M. P. (1992), *Riemannian Geometry*, Birkhauser Boston, MA.

Griffiths, T. L. & Steyvers, M. (2004), 'Finding scientific topics', *Proceedings of the National Academy of Sciences* **101**(suppl_1), 5228–5235.
**URL:** *https://www.pnas.org/doi/abs/10.1073/pnas.0307752101*

Hoffman, M., Blei, D. M., Wang, C. & Paisley, J. (2012), 'Stochastic variational inference'.
**URL:** *https://arxiv.org/abs/1206.7051*

Hutto, C. & Gilbert, E. (2014), 'Vader: A parsimonious rule-based model for sentiment analysis of social media text', *Proceedings of the International AAAI Conference on Web and Social Media* **8**(1), 216–225.
**URL:** *https://ojs.aaai.org/index.php/ICWSM/article/view/14550*

Jensen, J. L. W. V. (1906), 'Sur les fonctions convexes et les inegualites entre les valeurs Moyennes'.

Jordan, M., Ghahramani, Z., Jaakkola, T. & Saul, L. (1999), 'An introduction to variational methods for graphical models', *Machine Learning* **37**, 183–233.

Kotz, S., Balakrishnan, N. & Johnson, N. L. (2000), Continuous multivariate distributions, Vol. 1.

Kullback, S. & Leibler, R. A. (1951), 'On Information and Sufficiency', *The Annals of Mathematical Statistics* **22**(1), 79 – 86.
**URL:** *https://doi.org/10.1214/aoms/1177729694*

Liu, T., Zhang, N. L. & Chen, P. (2014), Hierarchical latent tree analysis for topic detection, *in* T. Calders, F. Esposito, E. Hullermeier & R. Meo, eds, 'Machine Learning and Knowledge Discovery in Databases', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 256–272.

Minka, T. & Lafferty, J. (2002), Expectation-propagation for the generative aspect model, *in* 'Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence', UAI'02, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p. 352â359.

Papadimitriou, C. H., Raghavan, P., Tamaki, H. & Vempala, S. (2000), 'Latent semantic indexing: A probabilistic analysis', *Journal of Computer and System Sciences* **61**(2), 217–235.
**URL:** *https://www.sciencedirect.com/science/article/pii/S0022000000917112*

Robbins, H. & Monro, S. (1951), 'A Stochastic Approximation Method', *The Annals of Mathematical Statistics* **22**(3), 400 – 407.
**URL:** *https://doi.org/10.1214/aoms/1177729586*

Rowels, S. & Hinton, G. (2002), Stochastic neighbour embedding, *in* 'Neural Information Processing Systems'.

van der Maaten, L. & Hinton, G. (2008), 'Visualizing data using t-sne', *Journal of Machine Learning Research* **9**(86), 2579–2605.
**URL:** *http://jmlr.org/papers/v9/vandermaaten08a.html*