Nonparametric Bayesian Topic Modelling with Auxiliary Data

Kar Wai Lim

A thesis submitted for the degree of
Doctor of Philosophy of
The Australian National University

July 2016

Except where otherwise indicated, this dissertation is my own original work.

Kar Wai Lim
24 July 2016

Supervisor

Wray Buntine
Professor, Monash University
Melbourne, VIC, Australia

Advisors

Edwin Bonilla
Senior Lecturer, The University of New South Wales
Sydney, NSW, Australia

Hanna Suominen
Senior Researcher, NICTA
Adjunct Assistant Professor, The Australian National University
Professional Associate, University of Canberra
Adjunct Professor, University of Turku (Finland)
Canberra ACT, Australia

Scott Sanner
Assistant Professor, Oregon State University
Oregon, OR, United States

To my family.

Acknowledgments

First, I would like to express my deepest gratitude to my supervisor, Wray Buntine, for the support and inspiration during my doctoral study. This dissertation would not be possible if not for his dedication and invaluable advice. I would also like to thank my advisors, Edwin Bonilla, Hanna Suominen, and Scott Sanner, for their comments in and out of research. In addition, I give thanks to the researchers for the interaction through visits and conferences.

My sincere thanks also goes to the Australian National University and NICTA for the financial support by way of a scholarship. In particular, I am grateful to Jochen Renz and Bob Williamson for supporting my application. I also appreciate the various assistance, encouragements, and moral supports from the university staff, NICTA employees, my peers, and my friends. Finally, I thank my family.

NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

Abstract

The intent of this dissertation in computer science is to study topic models for text analytics. The first objective of this dissertation is to incorporate auxiliary information present in text corpora to improve topic modelling for natural language processing (NLP) applications. The second objective of this dissertation is to extend existing topic models to employ state-of-the-art nonparametric Bayesian techniques for better modelling of text data. In particular, this dissertation focusses on:

- incorporating hashtags, mentions, emoticons, and target-opinion dependency present in tweets, together with an external sentiment lexicon, to perform opinion mining or sentiment analysis on products and services;
- leveraging abstracts, titles, authors, keywords, categorical labels, and the citation network to perform bibliographic analysis on research publications, using a supervised or semi-supervised topic model; and
- employing the hierarchical Pitman-Yor process (HPYP) and the Gaussian process (GP) to jointly model text, hashtags, authors, and the follower network in tweets for corpora exploration and summarisation.

In addition, we provide a framework for implementing arbitrary HPYP topic models to ease the development of our proposed topic models, made possible by modularising the Pitman-Yor processes. Through extensive experiments and qualitative assessment, we find that topic models fit better to the data as we utilise more auxiliary information and by employing the Bayesian nonparametric method.

Contents

List of Figures

List of Tables

List of Acronyms

AG	Adaptor Grammars
ASUM	Aspect and Sentiment Unification Model
ATM	Author-Topic Model
BUGS	Bayesian Inference Using Gibbs Sampling
CAT	Citation Author Topic
CNTM	Citation Network Topic Model
CRP	Chinese Restaurant Process
DP	Dirichlet Process
EM	Expectation-Maximisation
GEM	Griffiths-Engen-McCloskey
GP	Gaussian Process
HBC	Hierarchical Bayes Compiler
HDP	Hierarchical Dirichlet Process
HPYP	Hierarchical Pitman-Yor Process
IDF	Inverse Document Frequency
ILDA	Interdependent Latent Dirichlet Allocation
JAGS	Just Another Gibbs Sampler
LDA	Latent Dirichlet Allocation
LDA-DP	Latent Dirichlet Allocation with Dirichlet Prior Modified
LSI	Latent Semantic Indexing
MCMC	Markov Chain Monte Carlo
MG-LDA	Multi-grain Latent Dirichlet Allocation
MH	Metropolis-Hastings
ML	Machine Learning
NLP	Natural Language Processing
NMI	Normalised Mutual Information
PDD	Poisson-Dirichlet Distribution
pLSA	Probabilistic Latent Semantic Analysis

pLSI	Probabilistic Latent Semantic Indexing
PMI	Pointwise Mutual Information
PMTLM	Poisson Mixed-Topic Link Model
POS	Part-Of-Speech
PYP	Pitman-Yor Process
SCNTM	Supervised Citation Network Topic Model
TF	Term Frequency
TNTM	Twitter Network Topic Model
TOTM	Twitter Opinion Topic Model

Chapter 1

Introduction

We live in the information age. With the Internet, information can be obtained easily and almost instantly. This has changed the dynamic of information acquisition. For example, we can now (1) attain knowledge by visiting digital libraries, (2) be aware of the world by reading news online, (3) seek opinions from social media, and (4) engage in political debates *via* web forums. As technology advances, more information is created, to a point where it is infeasible for a person to digest *all* the available content. To illustrate, in the context of PubMed, a healthcare database, the number of entries has seen a growth rate of approximately 3,000 new entries per day in the ten-year period from 2003 to 2013 [Suominen *et al.*, 2014]. This motivates the use of machines to automatically organise, filter, summarise, and analyse the available data for the users. To this end, researchers have developed various methods, which can be broadly categorised into computer vision [Low, 1991; Mai, 2010], speech recognition [Rabiner and Juang, 1993; Jelinek, 1997], and *natural language processing* (NLP) [Manning and Schütze, 1999; Jurafsky and Martin, 2000]. This dissertation focuses on text analysis within NLP.

In *text analytics*, which is often associated with *text mining*, researchers seek to accomplish various goals, including *sentiment analysis* (or opinion mining) [Pang and Lee, 2008; Liu, 2012], *topic modelling* (or topic segmentation) [Blei, 2012], *information retrieval* [Manning *et al.*, 2008], and *text summarisation* [Lloret and Palomar, 2012]. To illustrate, sentiment analysis can be used to extract digestible summaries or reviews on products and services, which can be valuable to consumers. On the other hand, topic models attempt to discover abstract topics that are present in a collection of text documents. Note that text mining is often associated to the analysis of a large text collection. Since we do not limit our work to only dealing with large text corpora, we will say that this dissertation focusses on topic modelling rather than text mining.

Initially, topic models were developed for unstructured text. Topic models were inspired by the *latent semantic indexing* (LSI) [Landauer *et al.*, 2007] and its probabilistic variant, *probabilistic latent semantic indexing* (pLSI), also known as *probabilistic latent semantic analysis* (pLSA) [Hofmann, 1999]. Pioneered by Blei *et al.* [2003], the *latent Dirichlet allocation* (LDA) is a fully *Bayesian* extension of the pLSI, and can be considered the simplest Bayesian topic model. The LDA is then extended to many different types of topic models. Some of them are designed for specific applications [Wei and Croft, 2006; Mei *et al.*, 2007], some of them model the structure in the text [Blei and Lafferty, 2006; Du, 2012], while some incorporate extra information in their modelling [Ramage *et al.*, 2009; Jin *et al.*, 2011].

This dissertation will concentrate on topic models that take into account additional information. This information can be *auxiliary data* (or metadata) that accompany the text, such as keywords (or tags), dates, authors, and sources; or external resources like word lexicons. For example, on *Twitter*, a popular social media platform, its messages, known as *tweets*, are often associated with several metadata like location, time published, and the user who has written the tweet. This information can also be used. For instance, Kinsella *et al.* [2011] model tweets with location data, while Wang *et al.* [2011b] use hashtags for sentiment classification on tweets. On the other hand, many topic models have been designed to perform bibliographic analysis by using auxiliary information. Most notable of these is the author-topic model [Rosen-Zvi *et al.*, 2004], which, as its name suggests, incorporates authorship information. In addition to authorship, the Citation Author Topic model [Tu *et al.*, 2010] and the Author Cite Topic Model [Kataria *et al.*, 2011] make use of citations to model research publications. There are also topic models that employ external resources to improve modelling. For instance, He [2012] incorporates a sentiment lexicon as prior information into the LDA for a weakly supervised sentiment analysis.

Considering theory, recent advances in Bayesian methods have produced topic models that utilise *nonparametric* Bayesian priors. The most direct approach of these is simply replacing *Dirichlet distributions* in the LDA by *Dirichlet process* (DP) [Ferguson, 1973], resulting in the hierarchical Dirichlet process LDA (HDP-LDA) proposed by Teh *et al.* [2006]. One can further extend the topic models by using the *Pitman-Yor process* (PYP) [Ishwaran and James, 2001] that generalises the DP, this includes Sato and Nakagawa [2010], Du *et al.* [2012b], Lindsey *et al.* [2012], among others. Besides more flexible modelling, other advantages of employing the nonparametric Bayesian method on topic models is the ability to infer the number of clusters and to estimate topic prior probabilities from the data. Using PYPs also allows the modelling of power-law properties exhibited by natural languages [Goldwater *et al.*, 2005].

This dissertation, in the field of computer science, aims to develop full nonparametric Bayesian topic models that incorporate auxiliary information, with a goal of producing more accurate models that work well in tackling several applications like sentiment analysis and bibliographic study. As a by-product, we wish to encourage the use of state-of-the-art Bayesian techniques in topic modelling, as well as encourage the incorporation of different kinds of auxiliary information. We note that this dissertation is adapted and compiled from several publications and some unpublished work. In the next section, we provide a list of references to the published and submitted papers acquired throughout the doctoral studies.

1.1 List of Published and Submitted Papers

The following papers are accepted for publication in peer reviewed conference proceedings and journal, listed in reverse chronological order:

1. Lim, K. W., Buntine, W. L., Chen, C., and Du, L. (2016). Nonparametric Bayesian topic modelling with the hierarchical Pitman-Yor processes. *International Journal of Approximate Reasoning*, 1(1):1–40.

2. Lee, Y., Lim, K. W., and Ong, C. S. (2016). Hawkes processes with stochastic excitations. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of the 33rd International Conference on Machine Learning*, ICML 2016, pages 79–88.

3. Lim, K. W. and Buntine, W. L. (2016). Bibliographic analysis on research publications using authors, categorical labels and the citation network. *Machine Learning*, 103(2):185–213.

4. Lim, K. W. and Buntine, W. L. (2014). Bibliographic analysis with the Citation Network Topic Model. In Phung, D. and Li, H., editors, *Proceedings of the Sixth Asian Conference on Machine Learning*, ACML 2014, pages 142–158. Brookline, Massachusetts, USA. Microtome Publishing.

5. Lim, K. W. and Buntine, W. L. (2014). Twitter Opinion Topic Model: Extracting product opinions from tweets by leveraging hashtags and sentiment lexicon. In Li, J., Wang, X. S., Garofalakis, M. N., Soboroff, I., Suel, T., and Wang, M., editors, *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM 2014, pages 1319–1328. New York City, New York, USA. ACM.

6. Lim, K. W., Chen, C., and Buntine, W. L. (2013). Twitter-Network Topic Model: A full Bayesian treatment for social network and text modeling. In *Advances in Neural Information Processing Systems: Topic Models Workshop*, NIPS Workshop 2013, pages 1–5. Lake Tahoe, Nevada, USA.

7. Lim, K. W., Sanner, S., and Guo, S. (2012). On the mathematical relationship between expected n-call@k and the relevance *vs.* diversity trade-off. In Hersh, W. R., Callan, J., Maarek, Y., and Sanderson, M., editors, *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 2012, pages 1117–1118. New York City, New York, USA. ACM.

The following papers are currently under review:

1. Lim, K. W., Lee, Y., Hanlen, L., and Zhao, H. Simulation of multidimensional Hawkes with dissimilar decays. *Submitted to Asian Conference on Machine Learning*, ACML 2016, pages 1–16.

1.2 Major Contributions

We outline some major contributions of this dissertation:

1. **Framework for Bayesian topic modelling:** We present a modelling framework for nonparametric Bayesian topic models that employ the hierarchical Pitman-Yor processes (HPYPs). The novelty of this framework lies in the modularisation of the Pitman-Yor processes (PYPs), allowing us to implement HPYP topic models that are of arbitrary structure. The framework is inspired by BUGS (stands for Bayesian inference using Gibbs sampling) [Lunn *et al.*, 2000], that performs inference on arbitrary Bayesian models. However, BUGS does not extend to nonparametric Bayesian models. Several other tools for automatic inference, such as JAGS (stands for just another Gibbs sampler) [Plummer, 2003] and Infer.NET [Minka *et al.*, 2014], do not work for HPYP topic models.

   The above framework has been successfully applied to implement several topic models that will be discussed in this dissertation, like the HDP-LDA,[1] a nonparametric extension to the *author-topic model* (ATM), the proposed *Twitter Opinion Topic Model* (TOTM), the *Citation Network Topic Model* (CNTM), and the *Twitter Network Topic Model* (TNTM). Here we note that the network component of the CNTM and the TNTM is simply implemented on top of the framework, with little modification to the framework.

2. **Opinion mining and sentiment analysis:** We create a nonparametric Bayesian topic model to perform opinion mining on tweets. The proposed model, TOTM, leverages auxiliary metadata that are present in tweets, such as hashtags, mentions, emoticons, and strong sentiment words for sentiment analysis. As an extension to the *interdependent LDA* (ILDA) [Moghaddam and Ester, 2011], the TOTM models the target-opinion pairs that are extracted from tweets directly. As such, the TOTM is able to discover target specific opinions, which is neglected in existing approaches.

   Another novelty of this work is a new formulation for incorporating sentiment prior information into topic models using existing public sentiment lexicon. Although there are some existing work [He, 2012; Ding *et al.*, 2008; Taboada *et al.*, 2011] that uses sentiment lexicon for sentiment analysis, their approaches tend to be *ad hoc* or rule-based in nature. In contrast, our formulation follows a full Bayesian approach, and it learns and updates itself with the available data.

   In addition, we illustrate the usefulness of the TOTM with several qualitative analysis and applications that cannot be obtained with other topic models. This includes (1) an opinion analysis of specific target words, which are only made possible by modelling the target-opinion interaction directly, (2) an

[1] This includes the vanilla HDP-LDA and its bursty variant, as will be mentioned in Chapter 7.

aspect-based opinion comparison of major brands, and (3) an extract of contrastive opinions on certain products. We note that this work is published in Lim and Buntine [2014b].

3. **Bibliographic analysis:** Bibliographic analysis on research publications is always of interest to the research community. For this, we propose the CNTM that models text, the corresponding publication metadata, as well as the associated citation network. Modelling the citation network with a topic model leads to a complicated learning algorithm if we were to apply the standard *Markov chain Monte Carlo* (MCMC) theory naïvely. Our contribution in this work, apart from designing a full Bayesian topic model for bibliographic analysis, is that we propose a novel and efficient learning algorithm for the CNTM. The proposed algorithm introduces auxiliary parameters and uses the delta method approximation [Oehlert, 1992], to allow some parameters from the network component to be assimilated into the topic model component. Hence, this leads to a simpler learning algorithm for the full model.

   Moreover, we propose a method to incorporate supervision into the CNTM. This uses the categorical information that is available to the research publications. We demonstrate that incorporating supervision leads to improvement on document clustering. For applications, we use the CNTM for (1) corpora exploration by extracting research topics, (2) analysis of authors' research areas, and (3) a visualisation of the author-topic network. The work on bibliographic analysis is published in Lim and Buntine [2014a]. An extended version of this work will be available in Lim and Buntine [2016].

4. **Bayesian modelling on tweets:** We propose a fully Bayesian nonparametric topic model, named the TNTM, to jointly model the text content of tweets, their hashtags, their authors, and the corresponding followers network. The novelty in this work is that the TNTM utilises the HPYP to model the tweets, and the Gaussian process (GP) to model the network. Albeit slightly complicated, the TNTM is carefully designed to model tweets.[2] Contrary to some existing topic models that treat hashtags as labels (*e.g.*, Tsai [2011]), we model the hashtags as words that share tokens with text in the tweets. However, they are captured by a different variable. The complexity of the learning algorithm comes from the fact that each PYP in the model can have multiple parent PYPs and that the GP is not a conjugate, thus we develop a sampler that deals with the PYPs in vector form. The main contribution of this work is the holistic model for tweets.

   Through experiments, we show that jointly modelling the text content and the followers network leads to an improvement in model fitting, as compared to individual modelling of the text content and the followers network. This

[2] Our ablation studies show that each part in the model is important.

supports the argument that the more data the better. On the other hand, applying the TNTM for automatic topic labelling suggests that hashtags are also good labels for topics. This work is published in Lim *et al.* [2013, 2016].

Besides the major contributions mentioned above, in Section 3.2.4, we derive the posterior of a hierarchical Dirichlet model in which one of the intermediate Dirichlet distribution is integrated out. We show that this is a mixture of *Dirichlet-multinomial* distributions. The mixture is linked to the *Chinese Restaurant Process* (CRP) representation when we introduce auxiliary variables that select one of the mixture. This result is currently unpublished.

1.3 Dissertation Outline

This dissertation is outlined as follows. In Chapter 2, we briefly review the necessary background for Bayesian modelling. In particular, we introduce the terminologies and the basic concepts for Bayesian models. We then discuss some commonly used algorithms for approximate Bayesian inference. We focus on the MCMC method that will be used in this dissertation, the respective algorithms are the *Metropolis-Hastings* (MH) algorithm and the *Gibbs sampler*.

In Chapter 3, we move on to describe the probability distributions and stochastic processes that we will use in this dissertation. The univariate probability distributions that are mentioned are the Bernoulli distribution, the binomial distribution, and the beta distribution. We discuss the conjugacy of the beta-binomial distribution. We then detail the multivariate counterpart of the above mentioned distributions. Besides, we also present a hierarchical Dirichlet model that serves as a simple analogue to the HPYP used in the proposed topic models in later chapters. For stochastic processes, we outline the DP and the PYP, which are the building blocks for the following chapters.

Next, we discuss some commonly used Bayesian topic models in Chapter 4. The simplest of these is LDA. LDA is often extended to more complicated models, a nonparametric extension of LDA is the HDP-LDA. We also discuss topic models that incorporate metadata in their model. Examples are the ATM, the tag-topic model, and the supervised LDA. We also mention some notable and relevant topic models.

Chapter 5 details our topic modelling design and its implementation. We present a generic HPYP topic model that will be extended later. We detail its generation process, its model representation using the CRP metaphor, its *posterior likelihood*, and the inference procedure. We then outline some standard evaluations for topic models. The technical details on implementing the topic models are also presented. The discussion on Chapter 5 will be referred extensively by the later chapters.

We introduce the TOTM in Chapter 6 for opinion mining on tweets. The TOTM utilises hashtags, emoticons, and a sentiment lexicon for sentiment analysis. The

TOTM is extended from the generic HPYP topic model and thus the outline in Chapter 6 is similar to that of Chapter 5. In addition, we describe a procedure to incorporate sentiment lexicon as prior information into topic modelling, which leads to improvement in sentiment classification. Moreover, we discuss the steps to perform data cleaning and preprocessing, which are also relevant for Chapter 7 and 8. We then perform experiments to assess the TOTM and present qualitative results that are made possible with the TOTM. A diagnostic of the TOTM is also presented.

Chapter 7 and 8 follow the same structure as Chapter 6 so we outline the difference. In Chapter 7, we perform bibliographic analysis on research publications with the proposed CNTM. The CNTM is also an extension of the above HPYP topic model. The auxiliary information used by the CNTM includes authors, categories, keywords, abstracts, titles, and the citation network. We propose a novel inference algorithm for the CNTM, which combines the network component and the topic model component for efficient learning. Furthermore, we propose a method to incorporate supervision into the CNTM. Experiments show improvement on quantitative evaluations and sound qualitative results.

Finally, we propose the TNTM in Chapter 8. The TNTM models the authors, hashtags, and the followers network alongside tweets. Note that rather as labels, the hashtags are treated as words in the TNTM. As with the previous two models, the TNTM is also extended from the HPYP topic model. To model the network, we employ the use of the GP, which leads to a very flexible modelling of tweets. For inference, we propose an MH algorithm to jointly learn the topic model and the follower network. In the experiments, our ablation studies show that each component of the TNTM is important. We also demonstrate the quality of the TNTM in applications such as topic labelling and analysis of authors. Chapter 9 concludes.

1.4 A Note on Notation

Before moving to the next chapter, we discuss the notation philosophy used in this dissertation. We first note that the variables in each chapter are self-contained, that is, we do not carry on the definition of a variable to the next chapter unless explicitly stated. However, we try to keep the meaning of the variables consistent throughout this dissertation. For example, the α and β in this dissertation are hyperparameters, even though they might not be the same across the chapters.

Next, we would like to point out that a lower case symbol can represent both scalar and vector, it will be clear given the context. We use bold face capital letters to denote the set of all relevant lower case variables, for instance, $\mathbf{A} = \{a_1, \ldots, a_K\}$, and each $a_i = (a_{i1}, \ldots, a_{iN})$.

Chapter 2

Bayesian Analysis

We first review the necessary background that is relevant to this dissertation. This chapter focuses on the basics of the Bayesian method and we introduce the terminologies used later in this dissertation in Section 2.1. Then, in Section 2.2, we present some approximation techniques for Bayesian inference. We will particularly focus on the Markov chain Monte Carlo (MCMC) method as they are employed in this dissertation. Examples of the MCMC techniques include the Metropolis-Hastings (MH) algorithm and the Gibbs sampler.

2.1 Bayesian Modelling

A classical (frequentist) statistical model treats its parameters as unknown *constants*. These parameters need to be estimated by *estimators* that are usually obtained from techniques such as maximum likelihood estimation and method of moments matching.[3] An estimator is also a *statistic*, that is, it is a function of the observed data.

In contrast, a Bayesian model regards its unknown parameters as *random variables*, each of them having a prior distribution of its own. Inference on these parameters are based on their posterior distributions obtained *via* the Bayes' rule, conditional on the observed data.

An advantage of Bayesian inference over the classical approach is that we can incorporate our prior knowledge of the parameters into the model, whether the priors are from our own strong beliefs or based on previous experiences. Even when no prior information is available, we can let the priors to be "uninformative" or "vague", and let the data influence the posterior distributions.

This chapter serves as a refresher on important aspects in Bayesian analysis that are relevant to this dissertation. It is assumed that readers understand the basics of Bayesian methods hence the following discussion will be brief and concise. A

[3] The maximum likelihood estimators refers to the parameter values that maximise the model likelihood (or log likelihood), while the estimators from the moments matching method are obtained by matching the theoretical moments with the moment from the data.

comprehensive review of Bayesian approach can be found in the introductory text *Bayesian Data Analysis* by Gelman *et al.* [2013] and *Bayesian Theory* by Bernardo and Smith [1994].

2.1.1 A Simple Model

This dissertation introduces various Bayesian terminologies and concepts by way of a simple Bayesian model, given below:

$$(y \mid a, b) \sim p(y \mid a, b)\,, \tag{2.1}$$

$$(a \mid b) \sim p(a \mid b)\,, \tag{2.2}$$

$$b \sim p(b)\,, \tag{2.3}$$

where a, b and y are the *random variables* of the model, the notation $(y \mid a, b) \sim p(y \mid a, b)$ means the value of y follows a *probability distribution* $p(y \mid a, b)$ given a and b.

2.1.2 Priors and Posteriors

In this Bayesian model, a and b are unknown parameters, each having a *prior distribution*; whereas y corresponds to an observable variable. Using the Bayes' rule, the *joint posterior density* of a and b can be written as

$$p(a, b \mid y) = \frac{p(y \mid a, b) p(a, b)}{p(y)}\,, \tag{2.4}$$

where $p(y) = \iint p(y \mid a, b) p(a, b)\, da\, db$ is the *marginal probability distribution* of y, and $p(a, b) = p(a \mid b) p(b)$ is the *joint probability distribution* of a and b.

It is more common to write the joint posterior density up to a proportionality,

$$p(a, b \mid y) \propto p(y \mid a, b) p(a, b)\,. \tag{2.5}$$

This is because $p(y)$ is often hard to compute (due to the integration) and does not depend on the parameters a and b. Writing the joint posterior density in this *proportionality formula* allows us to avoid evaluating $p(y)$, but we are still able to analyse the posterior (*e.g.*, using the MCMC method, see Subsection 2.2).

If there is only one parameter of interest to be analysed, we can integrate out the other nuisance parameters to obtain the *marginal* posterior density. In this case, the marginal posterior densities for parameters a and b are

$$p(a \mid y) = \int p(a, b \mid y)\, \mathrm{d}b\,, \qquad p(b \mid y) = \int p(a, b \mid y)\, \mathrm{d}a\,. \tag{2.6}$$

2.1.3 Posterior Inferences

Unlike classical statistics, where inference on the unknown parameters are summarised into a single number and its confidence interval, the Bayesian approach enables us to analyse the distributions of the unknown parameters, that is, through the posterior distributions. Nevertheless, it is very useful to look at the key statistics of the posterior distributions, just like the classical approach.

The quantities of interest are posterior mean, median and mode, which are readily attainable from the posterior distributions. For confidence interval, a Bayesian equivalent would be the *central posterior density region* or the *highest posterior density region*, for details, see Jaynes and Kempthorne [1976].

In this example, assuming we have seen n values of y, namely $Y = (y_1, \dots, y_n)$, the joint posterior distribution can be derived as

$$p(a,b \,|\, Y) = \frac{p(Y \,|\, a,b)\, p(a,b)}{p(Y)} \propto p(Y \,|\, a,b)\, p(a,b) = p(a,b) \prod_{i=1}^{n} p(y_i \,|\, a,b)\,. \tag{2.7}$$

Here, we have used the fact that the observed variable y is independent and identically distributed (conditioned on the model parameters a and b). Having the joint posterior, the relevant marginal posterior distributions can then be derived in the usual way.

2.1.4 Predictive Inferences

In addition to obtaining inferences on the model parameters, an important use of Bayesian method is to perform prediction on future data, which is given more emphasis in practice. Performing predictive inference involves deriving the posterior distribution of the future values, conditioning on the observed data. Such a distribution is named the *posterior predictive distribution*.

For instance, say, we are interested in predicting a future value of y, denoted as $\tilde{y}$; the posterior predictive distribution of $\tilde{y}$ is

$$p(\tilde{y} \,|\, Y) = \iint p(\tilde{y}, a, b \,|\, Y)\, \mathrm{d}a\, \mathrm{d}b = \iint p(\tilde{y} \,|\, a,b)\, p(a,b \,|\, Y)\, \mathrm{d}a\, \mathrm{d}b\,, \tag{2.8}$$

noting that $\tilde{y}$ is conditionally independent of the data Y, conditioned on parameters a and b.

In principle, analysis of posterior distributions can be generalised to any quantity of interest related to the parameters in the model. For instance, we can analyse the posterior for any function of $\tilde{y}$, which has specific meaning in real world application.

2.2 Approximate Bayesian Inference

Performing Bayesian inference is essentially just analysing the marginal posterior distributions of quantity of interest (parameters, future data, and their functions). A standard approach involves deriving the posterior distributions using the Bayes' rule and then calculating the statistics of the posterior distributions (such as mean, variance, *etc.*). Often, deriving the marginal posterior distributions is extremely difficult, if not impossible, that is, there is no closed form solution for the posterior (though using a conjugate prior helps to alleviate the difficulty); this calls for special techniques to evaluate the posterior distributions, such as numerical integration, however, these can be tedious and time consuming.

Alternatively, MCMC methods are proposed to avoid the need to derive the required marginal posterior distributions [Gelman *et al.*, 2013]. MCMC methods allow us to sample quantities of interest from the posterior distributions directly and the relevant statistics can be computed from the samples. The merit of MCMC methods comes from the ease of implementing such methods. However, at the expense of longer computation time required to achieve good inference.

If a faster approximation is needed, variational Bayesian methods (or variational inference) [Bishop, 2006] are the next best substitute for MCMC methods. Variational methods can be seen as an extension of the expectation-maximisation (EM) algorithm [Dempster *et al.*, 1977], as they involve iterative updates of the parameters *via* E-steps and M-steps. Despite the increase speed in obtaining the inference, the drawback of using variational approaches is that deriving the needed equations requires great amount of work if not impossible.

This section provides a brief review of MCMC methods as they will be primarily used in the later chapters. Other methods are mentioned, but they are not the focus in this dissertation.

2.2.1 Markov Chain Monte Carlo Methods

The heart of a MCMC method lies in constructing a Markov chain of parameters for which their sampling distributions converge to the desired distributions (in this case the posterior distributions). Hence these samples can be treated as if they are drawn from the posterior distributions directly.

Most notable of the MCMC methods are the Metropolis-Hastings (MH) algorithms (also known as generalised Metropolis algorithms) [Metropolis *et al.*, 1953; Hastings, 1970] and Gibbs samplers [Geman and Geman, 1984]. They are discussed in Section 2.2.1.1 and Section 2.2.1.2.

2.2.1.1 Metropolis-Hasting Algorithm

Metropolis-Hasting algorithm was first proposed by Metropolis *et al.* [1953] and subsequently generalised by Hastings [1970]. The MH algorithm only requires that the joint posterior distribution of the model parameters, up to a proportionality constant, is known; note that the joint posterior distribution can be easily found using the Bayes' rule, which is proportional to the prior times likelihood.[4]

Let $\theta = (\theta_1, \ldots, \theta_k)$ represent a set of parameters having prior distributions given by $p(\theta)$ and $y = (y_1, \ldots, y_n)$ denotes the observed data, assuming the following simple Bayesian model:

$$(y \,|\, \theta) \sim p(y \,|\, \theta)\,, \tag{2.9}$$

$$(\theta) \sim p(\theta)\,, \tag{2.10}$$

then the joint posterior distribution is just

$$p(\theta \,|\, y) \propto p(y \,|\, \theta) p(\theta)\,. \tag{2.11}$$

Here, we are interested in the marginal posterior distribution $p(\theta_i \,|\, y)$ and the posterior predictive distribution $p(\tilde{y} \,|\, y)$ for future data $\tilde{y}$, the distribution of interest is known as the *target distribution*. In order to make inference on the quantities of interest, the MH algorithm creates a sequence of random values whose distributions converge to the target distributions. The MH algorithm can be summarised in Algorithm 2.1.

We note that the proposal distributions do not necessarily have to be dependent on any variables in the model. Also note that to accept the candidate value in Step 3(c) in Algorithm 2.1, we would need to generate a uniform random number u between 0 and 1 and accept the candidate value if $u < A'$.

With a large sample size R, the distributions of the random samples of θ can be said to converge in distribution to the marginal posterior distributions. To lessen the effect of a potentially badly chosen starting values that might not represent the samples of the posterior distributions, we remove first B sets of samples from the inference, so that the remaining $R - B$ samples are more appropriate in representing the posterior distributions. Here, B is named *burn-in* and we say the discarded samples are *burned*, or *burnt*.

A drawback of the MH algorithm is that the random values are not truly independent, their correlation comes from the Markov chain method where the next simulated value is obtained from its predecessor. To overcome this, only every other t-th samples (*e.g.*, $t = 5$) are used for the inference; this is called *thinning*. However, thinning reduces the quality of the inference when the sample size becomes smaller, or leads to a great increase in computational time in order to gather more samples.

[4] Likelihood here refers to the probability density function of the observed data.

Algorithm 2.1 Metropolis-Hasting algorithm

1. Pick initial values for $\theta = (\theta_1, \ldots, \theta_k)$, denote it $\theta^{(0)} = \left(\theta_1^{(0)}, \ldots, \theta_k^{(0)}\right)$, which will be the starting point for the Markov chain.

2. Define the *proposal distributions* (also known as the *jumping distributions*) for each parameter, $f(\theta_i^* | y, \theta)$, $i = 1, \ldots, k$, from which the next value for θ_i is sampled. The proposal distribution is usually chosen such that it is easy to sample and has roughly the same shape as the target distribution.

3. For $r = 1, \ldots, R$:
   For $j = 1, \ldots, k$:

   (a) Sample a candidate value, θ_j^* from the proposal distribution specified in Step 2 given the other values of θ:

   $$f\left(\theta_j^* \,\middle|\, y, \theta_1^{(r)}, \ldots, \theta_{j-1}^{(r)}, \theta_j^{(r-1)}, \theta_{j+1}^{(r-1)}, \ldots, \theta_k^{(r-1)}\right).$$

   (b) Calculate ratio of densities for θ_j^*, defined as

   $$\begin{aligned} A = {} & \frac{p\left(\theta_1^{(r)}, \ldots, \theta_{j-1}^{(r)}, \theta_j^*, \theta_{j+1}^{(r-1)}, \ldots, \theta_k^{(r-1)} \,\middle|\, y\right)}{p\left(\theta_1^{(r)}, \ldots, \theta_{j-1}^{(r)}, \theta_j^{(r-1)}, \theta_{j+1}^{(r-1)}, \ldots, \theta_k^{(r-1)} \,\middle|\, y\right)} \\ & \times \frac{f\left(\theta_j^{(r-1)} \,\middle|\, y, \theta_1^{(r)}, \ldots, \theta_{j-1}^{(r)}, \theta_j^*, \theta_{j+1}^{(r-1)}, \ldots, \theta_k^{(r-1)}\right)}{f\left(\theta_j^* \,\middle|\, y, \theta_1^{(r)}, \ldots, \theta_{j-1}^{(r)}, \theta_j^{(r-1)}, \theta_{j+1}^{(r-1)}, \ldots, \theta_k^{(r-1)}\right)}. \end{aligned}$$

   (c) Update the value of $\theta_j^{(r)}$ to θ_j^* with acceptance probability $A' = \min(A, 1)$. If θ_j^* is not accepted, then set $\theta_j^{(r)} = \theta_j^{(r-1)}$, that is, the next value for θ_j retains the same value.

Note that to make inference on the future data $\tilde{y}$, we simply generate a sample of $\tilde{y}$ using the simulated parameters θ, the generated $\tilde{y}$ will be distributed approximately from the posterior predictive distribution. This method also applies to any function of the parameters or any random variable conditioned on the parameters.

2.2.1.2 Gibbs Sampling

Gibbs sampling is a special case of the MH algorithm, for which the proposal distributions take a particular form [Geman and Geman, 1984]. More specifically, the proposal distribution of each parameter is a conditional posterior distribution, given the data and all other parameters (except itself):

$$f(\theta_i^* \,|\, y, \theta) = p(\theta_i^* \,|\, y, \theta_{-i}) \,, \tag{2.12}$$

where $\theta_{-i} = (\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_k)$ is a set of all parameters except for θ_i.

With this specification of proposal distribution, the acceptance probabilities will be equal to 1; denoting $\theta_{-i}^{(r-1)} = \left(\theta_1^{(r)}, \ldots, \theta_{i-1}^{(r)}, \theta_{i+1}^{(r-1)}, \ldots, \theta_k^{(r-1)}\right)$, the acceptance probability for candidate value θ_i^* is derived as

$$\begin{aligned} A(\theta_i^*) &= \frac{p\left(\theta_1^{(r)}, \ldots, \theta_{i-1}^{t}, \theta_i^*, \theta_{i+1}^{(r-1)}, \ldots, \theta_k^{(r-1)} \,\middle|\, y\right)}{p\left(\theta_1^{(r)}, \ldots, \theta_{i-1}^{(r)}, \theta_i^{(r-1)}, \theta_{i+1}^{(r-1)}, \ldots, \theta_k^{(r-1)} \,\middle|\, y\right)} \\ &\quad \times \frac{p\left(\theta_i^{(r-1)} \,\middle|\, y, \theta_1^{(r)}, \ldots, \theta_{i-1}^{(r)}, \theta_{i+1}^{(r-1)}, \ldots, \theta_k^{(r-1)}\right)}{p\left(\theta_i^* \,\middle|\, y, \theta_1^{(r)}, \ldots, \theta_{i-1}^{(r)}, \theta_{i+1}^{(r-1)}, \ldots, \theta_k^{(r-1)}\right)} \\ &= \frac{p\left(\theta_i^* \,\middle|\, y, \theta_{-i}^{(r-1)}\right) p\left(\theta_{-i}^{(r-1)} \,\middle|\, y\right)}{p\left(\theta_i^{(r-1)} \,\middle|\, y, \theta_{-i}^{(r-1)}\right) p\left(\theta_{-i}^{(r-1)} \,\middle|\, y\right)} \; \frac{p\left(\theta_i^{(r-1)} \,\middle|\, y, \theta_{-i}^{(r-1)}\right)}{p\left(\theta_i^* \,\middle|\, y, \theta_{-i}^{(r-1)}\right)} \\ &= 1 \,. \end{aligned} \tag{2.13}$$

Hence under Gibbs sampling, all candidate values are accepted. The Gibbs sampler is preferred to the MH algorithm because it produces no *wastage* (no candidate value is rejected). However, sometimes a considerable amount of effort is needed to derive the distribution of the conditional density and/or to sample from it. Thus there may be a trade-off between efficiency and simplicity.

Note that it is not necessary for us to sample each parameter sequentially (as described above), one can develops an algorithm that updates more than one parameter at once in each iteration; such MCMC samplers are called *blocked Gibbs sampler* [Liu, 1994]. Also, in practice we are usually only interested in a certain subset of the parameters and do not care about the others; in such cases we can derive a *collapsed Gibbs samplers* [Liu, 1994] for which the nuisance parameters are integrated out, doing this requires more effort in derivation but the sampler would be much more efficient.

2.2.2 Other Methods

Another popular methods for approximate Bayesian inference are the variational inference [Bishop, 2006] and a stochastic version of the variational inference [Hoffman *et al.*, 2013]. The variational inference techniques approximate the posterior distributions with a tractable distribution family, usually of the Gaussian distributions, and minimises the Kullback-Leibler divergence [Kullback and Leibler, 1951] to obtain point estimates of the parameters. Although this approach is faster, it often suffers from being stuck at the local optima. Derivation of its algorithm is also non-trivial.

Other methods for approximate Bayesian inference includes Expectation Propagation [Minka, 2001] and the expectation-maximisation (EM) algorithm [Dempster *et al.*, 1977]. These approaches will not be discussed in this dissertation and we refer the interested readers to the references thereof.

2.3 Summary

This chapter reviews some basic of Bayesian methods, including how a Bayesian model is constructed and the how to make inference on quantities of interest in the model. Due to the difficulty to make inference on posterior analytically, which is usually complex in practical situations, various approximation approaches were reviewed; emphasis was given in the discussion of Markov chain Monte Carlo methods as these will be used primarily in this dissertation.

In the next chapter, we continue with the discussion on some important probability distributions and stochastic processes that are used in this dissertation. In particular, we note that they are discussed in the framework of Bayesian modelling.

Chapter 3

Probability Distributions and Stochastic Processes

This chapter provides a brief review on probability distributions and stochastic processes. The following illustrated probability distributions and stochastic processes are chosen on the basis of relevance to this dissertation; they are only a tiny portion of all existing (and important) distributions, see Walck [2007] for a comprehensive list of other important probability distributions. We first describe some simple probability distributions in Sections 3.1 and 3.2. Section 3.3 describes the nonparametric approach in Bayesian methods and mentions some stochastic processes.

3.1 Univariate Probability Distributions

We first discuss the simple univariate probability distributions. These distributions are characterised by the fact that they generate one variable at a time.

3.1.1 Bernoulli Distribution

The *Bernoulli distribution* can be considered as the simplest of all distributions. It is a *discrete* distribution (*i.e.*, the outcome takes on a fixed value) with only two outcomes: 0 and 1. A classical example having such distribution would be the number of heads obtained from a *single* toss of a bent coin.

Let θ denote the probability of landing a head, and x denotes the number of heads obtained, we say x follows a Bernoulli distribution with parameter θ, which can be presented as follows:

$$(x \mid \theta) \sim \text{Bernoulli}(\theta)\,. \tag{3.1}$$

The *probability density function*[5] associated with x is given by

$$p(x \mid \theta) = \theta^x (1-\theta)^{1-x}\,, \qquad\qquad x \in \{0,1\},\;\; \theta \in [0,1]\,. \tag{3.2}$$

[5] It should be called the probability mass function in the case of a discrete probability distribution, but for convenience, we call it a probability density function as in the continuous case.

3.1.2 Binomial Distribution

The binomial distribution is a generalisation of the Bernoulli distribution with multiple trials. Following the above example, if we throw the same bent coin n times and again denote x as the number of heads obtained, then x follows a binomial distribution with parameter n and θ:

$$(x \mid n, \theta) \sim \text{Binomial}(n, \theta)\,. \tag{3.3}$$

As with the Bernoulli distribution, it is a discrete distribution, but now with $(n+1)$ outcomes from n trials. The probability density for x is given as

$$p(x \mid n, \theta) = \binom{n}{x} \theta^{x} \, (1-\theta)^{n-x}\,, \qquad x \in \{0, 1, \ldots, n\},\ \ \theta \in [0,1]\,, \tag{3.4}$$

where the notation $\binom{n}{x}$ denotes the binomial coefficient, given as

$$\binom{n}{x} = \frac{n!}{x!\,(n-x)!}\,. \tag{3.5}$$

3.1.3 Beta Distribution

In contrast to the Bernoulli and the binomial distribution, the beta distribution is a *continuous* distribution (*i.e.*, the outcome can be any real number) for which the outcome can take values between 0 and 1 (inclusive). The beta distribution is usually used as a prior distribution for the probability of an event. For example, we can model the probability of getting a head from a coin toss, θ, by a beta distribution:

$$p(\theta \mid a, b) = \frac{1}{B(a,b)} (\theta)^{a-1} (1-\theta)^{b-1}\,, \qquad \theta \in [0,1],\ \ a > 0,\ \ b > 0\,. \tag{3.6}$$

Here, the parameters a and b are known as shape parameters, and $B(\cdot,\cdot)$ is called the beta function, which serves as a normalisation constant. The beta function can also be written as a product of gamma functions:

$$B(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}\,. \tag{3.7}$$

Note that the beta distribution is a *conjugate* distribution of the binomial distribution (and also of the Bernoulli distribution). This means that the prior and posterior distributions of θ will be of the same family of distributions, namely the beta family. This convenient property also allows a tractable derivation of a *compound distribution* named the beta-binomial distribution.

3.1.4 Beta-Binomial Distribution

Consider the following Bayesian model:

$$(x \mid n, \theta) \sim \text{Binomial}(n, \theta)\,, \tag{3.8}$$

$$(\theta \mid a, b) \sim \text{Beta}(a, b)\,. \tag{3.9}$$

where a and b are *hyperparameters* associated with prior θ. Note that the variables n, a and b are known (or chosen to be certain values) in the model.

It is not difficult to show that the posterior of θ follows a beta distribution:

$$\begin{aligned} p(\theta \mid x, n, a, b) &\propto p(x \mid n, \theta)\, p(\theta \mid a, b) \\ &\propto \theta^{a+x-1}(1-\theta)^{b+n-x-1}\,, \end{aligned} \tag{3.10}$$

that is, $(\theta \mid x, n, a, b) \sim Beta(a + x, b + n - x)$. Often times, we rewrite Equation (3.10) as $p(\theta \mid x)$, implicitly conditioning on known variables (n, a and b) for simplicity and ease of reading. This conjugacy also enables us to analytically derive the compound distribution of x by integrating out the parameter θ:

$$\begin{aligned} p(x \mid n, a, b) &= \int_0^1 p(x \mid n, \theta)\, p(\theta \mid a, b)\, \mathrm{d}\theta \\ &= \binom{n}{x} \frac{1}{B(a,b)} \int_0^1 \theta^{a+x-1}(1-\theta)^{b+n-x-1}\, \mathrm{d}\theta \\ &= \binom{n}{x} \frac{B(a+x, b+n-x)}{B(a,b)}\,. \end{aligned} \tag{3.11}$$

This distribution is known as the beta-binomial distribution. Note that the integral in Equation (3.11) is easily computed by recognising that it is part of the posterior distribution of θ.

For situations where there is *a priori* ignorance regarding θ (*i.e.*, we do not know what a and b are), three specifications have been proposed: *uniform* prior ($a = b = 1$), *improper* prior[6] ($a = b = 0$) and Jeffreys prior ($a = b = 1/2$). Each of these has its advantages and disadvantages. However, given large sample size (often true for computer science application), the differences between using the three priors tend to be negligible.

Note that the improper prior is not a proper probability distribution in which the density does not sum up (or integrate) to 1. When one uses an improper prior, care must be taken to ensure that the posterior distribution is proper, otherwise the inference obtained is completely useless!

[6] Which is also known as the Haldane prior [Haldane, 1932].

3.2 Multivariate Probability Distributions

Multivariate probability distributions are a generalisation of univariate probability distributions discussed in Section 3.1. The outcomes from a multivariate distribution spans multiple dimensions and their values are often dependent on one another.

As above, this subsection reviews some multivariate distributions that are relevant to this dissertation. Again, we refer the readers to Walck [2007] for more information on these distributions and details on the other distributions.

3.2.1 Multinomial Distribution

The multinomial distribution is a multivariate generalisation of the binomial distribution. While each binomial trial relates to an event being success (1) or failure (0), a trial in multinomial distribution results in a success in exactly one of k possible outcomes. For example, rolling a die. A sample from a multinomial distribution consists of the frequency of the successes in each outcome after n trials.

Instead of having a single parameter on probability of success (like θ in the binomial distribution), the multinomial distribution requires parameters in the form of a probability vector (length k), which comprises of the probability of getting a success in each outcome. We denote this probability vector as $\theta = (\theta_1, \ldots, \theta_k)$.

Let $x = (x_1, \ldots, x_k)$ be a vector of frequencies correspond to successes in each outcome after n rolls of a k-sided die (does not need to be a fair die) with the probability of success in each outcome (rolling 1 to k) defined by $\theta = (\theta_1, \ldots, \theta_k)$, then the probability density function of x is

$$p(x \,|\, n, \theta) = \binom{n}{x} (\theta_1)^{x_1} \ldots (\theta_k)^{x_k}, \qquad x_i \in \{0, \ldots, n\}, \;\; \theta_i \in [0, 1]. \tag{3.12}$$

Note that the constraints $\sum_{i=1}^{k} x_i = n$ and $\sum_{i=1}^{k} \theta_i = 1$ need to be satisfied.

The multinomial distribution is equivalent to the categorical distribution or simply the 'discrete distribution' when n is equal to 1, but the sample space is now one of the possibility (out of k) rather than counts. On the other hand, the multinomial distribution reduces to the binomial distribution when $k = 2$.

3.2.2 Dirichlet Distribution

The Dirichlet distribution is a multivariate generalisation of the beta distribution. As with the beta distribution, the Dirichlet distribution is often used as a prior distribution for a probability vector representing probabilities of *mutually exclusive* events, such as the probability distribution of a die roll. The Dirichlet distribution is conjugate to the multinomial distribution, exactly like the relationship between the beta distribution and the binomial distribution.

The Dirichlet distribution is parameterised by a vector $\alpha = (\alpha_1, \ldots, \alpha_k)$ of length k, and has the following probability density function:

$$p(\theta \,|\, \alpha) = \frac{1}{B_k(\alpha)} \, (\theta_1)^{\alpha_1 - 1} \ldots (\theta_k)^{\alpha_k - 1} \,, \qquad \theta_i \in [0,1], \;\; \alpha_i > 0 \,, \tag{3.13}$$

where $\sum_{i=1}^{k} \theta_i = 1$; $B_k(\alpha)$ is a k-dimensional generalisation of the beta function that normalises the distribution, defined as

$$B_k(\alpha) = \frac{\Gamma(\alpha_1) \ldots \Gamma(\alpha_k)}{\Gamma(\alpha_1 + \cdots + \alpha_k)} \,. \tag{3.14}$$

The Dirichlet distribution can also be parameterised by its mean μ and precision ρ, where $\rho = \sum_{i=1}^{k} \alpha_i$ and $\mu = \alpha / \rho$. This parameterisation is sometimes preferred due to its greater interpretability. Like the multinomial distribution, the Dirichlet distribution reduces to the beta distribution when $k = 2$.

3.2.3 Dirichlet-Multinomial Distribution

Due to the conjugacy of the Dirichlet distribution to the multinomial distribution, a compound distribution named the Dirichlet-multinomial distribution can be constructed similarly to the construction of the beta-binomial distribution. Specifically, the distribution arises from the following Bayesian model:

$$(x \,|\, n, \theta) \sim \text{Multinomial}(n, \theta) \,, \tag{3.15}$$

$$(\theta \,|\, \alpha) \sim \text{Dirichlet}(\alpha) \,. \tag{3.16}$$

Again, we can show that the posterior of θ follows a Dirichlet distribution:

$$(\theta | x, n, \alpha) \sim \text{Dirichlet}(\alpha + x). \tag{3.17}$$

The parameters $\alpha = (\alpha_1, \ldots, \alpha_k)$ are also known as pseudocounts as they are added to the observed outcomes. Note that the posterior is simply an empirical distribution of the observed outcomes when an improper prior (where all $\alpha_i = 0$) is used.

The probability density function of Dirichlet-multinomial distribution — where the Dirichlet parameter is integrated out — can be derived as

$$\begin{aligned} p(x \,|\, n, \alpha) = \int p(x \,|\, n, \theta) \, p(\theta \,|\, \alpha) \, \mathrm{d}\theta &= \binom{n}{x} \frac{1}{B_k(\alpha)} \int (\theta_1)^{\alpha_1 + x_1 - 1} \ldots (\theta_k)^{\alpha_k + x_k - 1} \, \mathrm{d}\theta \\ &= \binom{n}{x} \frac{B_k(\alpha + x)}{B_k(\alpha)} \,. \end{aligned} \tag{3.18}$$

We note that if an array of discrete distributions of length n is used instead of the multinomial distribution, that is, $(z_i \,|\, \theta) \sim \text{Discrete}(\theta)$ for $i = 1, \dots, n$, then

$$p(z \,|\, n, \alpha) = \frac{B_k(\alpha + x)}{B_k(\alpha)} \,. \tag{3.19}$$

The $\binom{n}{x}$ term is dropped since the sample space is now consist of all permutations of z instead of counts.

3.2.4 Hierarchical Dirichlet Model

Consider the following hierarchical Bayesian model in which the prior for a Dirichlet distribution is also Dirichlet distributed:

$$(x \,|\, n, \theta) \sim \text{Multinomial}(n, \theta) \,, \tag{3.20}$$

$$(\theta \,|\, \alpha) \sim \text{Dirichlet}(c\alpha) \,, \tag{3.21}$$

$$(\alpha \,|\, \beta) \sim \text{Dirichlet}(\beta) \,. \tag{3.22}$$

Here, c is an arbitrary positive constant and β is a positive (non-zero) vector.

From Equation (3.18), we know that the Dirichlet parameter θ can be integrated out, but can we also integrate out the Dirichlet parameter α? Here we show that all Dirichlet parameters in a hierarchical model can be integrated out *via* the following derivation.

$$\begin{aligned} p(x \,|\, n, \beta) &= \int p(x \,|\, n, \alpha) \, p(\alpha \,|\, \beta) \, \mathrm{d}\alpha \\ &= \binom{n}{x} \int \frac{B_k(c\alpha + x)}{B_k(c\alpha)} \, \frac{1}{B_k(\beta)} \prod_{i=1}^{k} (\alpha_i)^{\beta_i - 1} \, \mathrm{d}\alpha \,. \end{aligned} \tag{3.23}$$

We can simplify the ratio of the beta functions using the fact that x is a vector of integers, as follows:

$$\begin{aligned} \frac{B_k(c\alpha + x)}{B_k(c\alpha)} &= \frac{\prod_{i=1}^{k} \Gamma(c\alpha_i + x_i)}{\Gamma(\sum_{i=1}^{k} c\alpha_i + x_i)} \, \frac{\Gamma(\sum_{i=1}^{k} c\alpha_i)}{\prod_{i=1}^{k} \Gamma(c\alpha_i)} \\ &= \frac{\Gamma(c)}{\Gamma(c+n)} \prod_{i=1}^{k} (\alpha_i)(\alpha_i + 1) \cdots (\alpha_i + x_i - 1) \\ &= \frac{\Gamma(c)}{\Gamma(c+n)} \prod_{i=1}^{k} \left(\sum_{j=1}^{x_i} S_j^{x_i} (\alpha_i)^j \right) \,, \end{aligned} \tag{3.24}$$

where the last line of Equation (3.24) is derived by expanding the inner products, here S_a^b denotes the generalised Stirling Numbers of the first kind [Buntine and Hutter, 2012, Theorem 17]. Replacing this formulation into Equation (3.23) gives

$$\begin{aligned} p(x \mid n, \beta) &= \binom{n}{x} \frac{\Gamma(c)}{\Gamma(c+n)} \sum_{t_1, \ldots, t_k} \left(\prod_{i=1}^{k} S_{t_i}^{x_i} \right) \frac{1}{B_k(\beta)} \int \prod_{i=1}^{k} (\alpha_i)^{\beta_i + t_k - 1} \, d\alpha \\ &= \binom{n}{x} \frac{\Gamma(c)}{\Gamma(c+n)} \sum_{t_1, \ldots, t_k} \frac{B_k(\beta + t)}{B_k(\beta)} \prod_{i=1}^{k} S_{t_i}^{x_i} \, . \end{aligned} \tag{3.25}$$

We note that t_i is strictly positive when x_i is non-zero, and the t_i can take values from 1 to x_i (inclusive).

From Equation (3.25), we can see that when the Dirichlet parameters are integrated out more than once, the distribution corresponds to mixtures of Dirichlet-multinomial distributions. For topic modelling, as we will discuss in later chapters, we can introduce an auxiliary variables called table counts to avoid dealing with the mixtures explicitly. The table counts can be viewed as indicators that select one of the mixture in Equation (3.25). Note that this is consistent with the Chinese Restaurant Process representation that will be discussed in Section 5.3.

Finally, this method can be applied recursively to a deeper hierarchical model of Dirichlet distributions. This gives a more complicated mixtures of Dirichlet-multinomial distributions.

3.3 Stochastic Processes and the Nonparametric Model

A *stochastic process* can be viewed as an extension of probability distribution, as it is a collection of *random variables* each having a probability distribution (which is related to one another). A stochastic process is usually used to represent the evolution of a system (*e.g.*, see Markov chain).

The term *nonparametric* model [Hjort *et al.*, 2010] has two meanings, the first refers to a model that contains no parameters at all, which assumes that data that are observed do not follow a given probability distribution; while the second refers to a model that does not assume a particular structure (*i.e.*, fixed probability distribution), the parameters in this model usually grow in size with the amount of data. In the Bayesian context, the term nonparametric refers to the latter.

This section covers some nonparametric stochastic processes that are related to this dissertation. A review on other stochastic processes is available in Çinlar [2011].

3.3.1 Dirichlet Process

The *Dirichlet process* (DP) is a stochastic process that can be thought of as an infinite-dimensional generalisation of the Dirichlet distribution. Unlike simple probability distributions, the DP is 'parameterised' by a probability distribution (named the *base distribution* or *base measure*) and a positive real number (the concentration parameter).

A sample from a DP is a probability distribution known as the *output distribution*. The support of the output distribution is the same as the base distribution, meaning that a sample drawn from the output distribution must also be possible to be drawn from the base distribution. As with the Dirichlet distribution and beta distribution, the DP is usually used as a prior in a hierarchical Bayesian model.

The DP is formally introduced by Ferguson [1973]. Formally, let H be a random measure on measurable space $(\mathcal{X}, \mathcal{B})$ and let β to be a positive real number. G is said to be a DP on $(\mathcal{X}, \mathcal{B})$ with a base measure H and a concentration parameter β if for any measurable partition $(A_1, \dots, A_k)$ of $\mathcal{X}$, the distribution of $\big(G(A_1), \dots, G(A_k)\big)$ is Dirichlet distributed with parameter $\big(\beta H(A_1), \dots, \beta H(A_k)\big)$:

$$G \sim \mathrm{DP}(\beta, H)\,, \tag{3.26}$$

then

$$\big(G(A_1), \dots, G(A_k)\big) \sim \mathrm{Dirichlet}\big(\beta H(A_1), \dots, \beta H(A_k)\big)\,. \tag{3.27}$$

From this definition, if H is a discrete probability distribution (over a finite space), then the DP is a Dirichlet distribution:

$$\mathrm{DP}(\beta, H) \equiv \mathrm{Dirichlet}(\beta H)\,, \tag{3.28}$$

where $H = (h_1, \dots, h_k)$ representing a probability vector.

When H is *non-discrete* (non-atomic, or continuous), a DP is essentially an infinite-dimensional Dirichlet distribution. To draw a sample from the DP, certain sampling schemes have been proposed, such as the *stick-breaking process* [Sethuraman, 1991] and the *Chinese restaurant process* (CRP), which is also known as the Blackwell-Macqueen urn scheme [Blackwell and MacQueen, 1973].

Note that a sample drawn from a DP is always discrete (this does not mean finite) even when the base distribution is continuous. Since in most real world applications a sample from a DP is finite given limited observations, we can treat a DP as a Dirichlet distribution, though using a DP allows us to model an unconstrained (and changeable) number of state space.

3.3.2 Pitman-Yor Process

The *Pitman-Yor process* (PYP) [Ishwaran and James, 2001] is also known as the two-parameter *Poisson-Dirichlet process*. The PYP is a two-parameter generalisation of the DP, now with an extra parameter α named the *discount parameter* in addition to the concentration parameter β. Similar to DP, a sample from PYP corresponds to a discrete distribution with the same support as its base distribution H. The underlying distribution of PYP is the *Poisson-Dirichlet distribution* (PDD), which was introduced by Pitman and Yor [1997].

The PDD is defined by its construction process. For $0 \leq \alpha < 1$ and $\beta > -\alpha$, let V_k be distributed independently as follows:

$$(V_k \,|\, \alpha, \beta) \sim \text{Beta}(1-\alpha, \beta + k\alpha)\,, \qquad \text{for } k = 1, 2, 3, \ldots\,, \tag{3.29}$$

and define $(p_1, p_2, p_3, \ldots)$ as

$$p_1 = V_1\,, \tag{3.30}$$

$$p_k = V_k \prod_{i=1}^{k-1} (1 - V_i)\,, \qquad \text{for } k \geq 2\,. \tag{3.31}$$

If we let $p = (\tilde{p}_1, \tilde{p}_2, \tilde{p}_3, \ldots)$ be a sorted version of $(p_1, p_2, p_3, \ldots)$ in descending order, then p is Poisson-Dirichlet distributed with parameter α and β:

$$p \sim \text{PDD}(\alpha, \beta)\,. \tag{3.32}$$

Note that the unsorted version $(p_1, p_2, p_3, \ldots)$ follows a $\text{GEM}(\alpha, \beta)$ distribution, which is named after Griffiths, Engen and McCloskey [Pitman, 2006].

With the PDD defined, we can then define the PYP formally. Let H be a distribution over a measurable space $(\mathcal{X}, \mathcal{B})$, for $0 \leq \alpha < 1$ and $\beta > -\alpha$, suppose that $p = (p_1, p_2, p_3, \ldots)$ follows a PDD (or GEM) with parameters α and β, then PYP is given by the formula

$$p(x \,|\, \alpha, \beta, H) = \sum_{k=1}^{\infty} p_k \, \delta_{X_k}(x)\,, \qquad \text{for } k = 1, 2, 3, \ldots\,, \tag{3.33}$$

where X_k are independent samples drawn from the base measure H and $\delta_{X_k}(x)$ represents probability point mass concentrated at X_k (*i.e.*, it is an indicator function that is equal to 1 when $x = X_k$ and 0 otherwise):

$$\delta_x(y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise}\ . \end{cases} \tag{3.34}$$

This construction is named the stick-breaking process. The PYP can also be constructed using an analogue to Chinese restaurant process (which explicitly draws a sequence of samples from the base distribution). A more extensive review on the PYP is given by Buntine and Hutter [2012].

For some applications such as natural language processing, a PYP is more suitable than a DP as it exhibits a power-law behaviour (when $\alpha \neq 0$), which is observed in natural languages [Goldwater *et al.*, 2005; Teh and Jordan, 2010]. Note that when the discount parameter α is 0, the PYP simply reduces to a DP.

3.3.3 Pitman-Yor Process with a Mixture Base

Note that the base measure H of a PYP is not necessarily restricted to a single probability distribution, we can let H be a mixture distribution such as

$$H = \rho_1 H_1 + \rho_2 H_2 + \cdots + \rho_n H_n \,, \tag{3.35}$$

where $\sum_{i=1}^{n} \rho_i = 1$ and $\{H_1, \ldots H_n\}$ is a set of distributions over the same measurable space $(\mathcal{X}, \mathcal{B})$ as H.

With this specification of H, the PYP is also named the compound Poisson-Dirichlet process in Du [2012], or the doubly hierarchical Pitman-Yor process in Wood and Teh [2009]. A special case of this is the DP equivalent, which is also known as the DP with mixed random measures in Kim *et al.* [2012]. We note that in the CRP representation, if the base distribution is a mixture of multiple PYP, we can treat the PYP to have multiple parent restaurants. More on this in Chapter 5.

Note that in the above discussion we have assumed constant values for the ρ_i, though of course we can go fully Bayesian and assign a prior distribution for each of them, a natural prior would be the Dirichlet distribution:

$$(\rho \,|\, \gamma) \sim \text{Dirichlet}(\gamma) \,, \tag{3.36}$$

where we defined $\rho = (\rho_1, \ldots, \rho_n)$ and $\gamma = (\gamma_1, \ldots, \gamma_n)$.

3.4 Summary

This chapter provides a brief review on some relevant and important probability distributions and their characteristics. In particular, we touch on the aspect of choosing conjugate priors to simplify the corresponding posterior distributions. This also led to the discussion on the Hierarchical Dirichlet Model in Section 3.2.4, which serves as a bridge to the discussion of some related stochastic processes.

An application of these probability distributions and stochastic processes is in the area of topic modelling. This will be reviewed in the next chapter.

Chapter 4

Topic Models

One example out of many successful Bayesian applications is topic modelling, which is an algorithm that automatically discovers the *latent* (or hidden) structure of a corpus of documents. Here, a document is not restricted to just text, it can be an image, video or even genes (with genetic information); essentially, topic modelling can be applied to any data that can be represented by a set of items/features [Blei *et al.*, 2003; Fergus *et al.*, 2005; Zheng *et al.*, 2006; Hospedales *et al.*, 2012]. In this dissertation, we discuss topic modelling in the context of text analysis.

"Topic modelling algorithms are statistical methods that analyse the words of the original texts to discover the themes that run through them, how those themes are connected to each other, and how they change over time" [Blei, 2012]. With topic models, we are able to analyse and summarise electronic documents — which are growing in size exponentially — quickly and automatically.

A *topic* is essentially a set of words grouped together by their co-occurrence and other factors (this depends on the topic model). Although a topic does not have a word or a title that describes itself, practitioners tend to represent a topic by the first *n* most significant words. To overcome this manual task, the research community has proposed several methods to label the topics autonomously. Recent attempts on automatic topic labelling include the work of Lau *et al.* [2011], Mao *et al.* [2012], Aletras and Stevenson [2014], and Cano Basave *et al.* [2014].

Topic modelling is being used in many domains such as text analysis and computer vision. In text analysis, topic modelling has been used for document clustering, topic exploration, sentiment analysis, text summarisation, document segmentation, and information retrieval [Blei, 2012]. In computer vision, topic modelling is successfully used in face recognition [Lu *et al.*, 2003] and scene recognition [Fei-Fei and Perona, 2005]. In this chapter, we discuss some popular topic models used in practice.

4.1 Latent Dirichlet Allocation

The *Latent Dirichlet allocation* (LDA) [Blei *et al.*, 2003] is the simplest Bayesian topic model; it is a fully Bayesian extension of the *probabilistic latent semantic indexing* (pLSI)

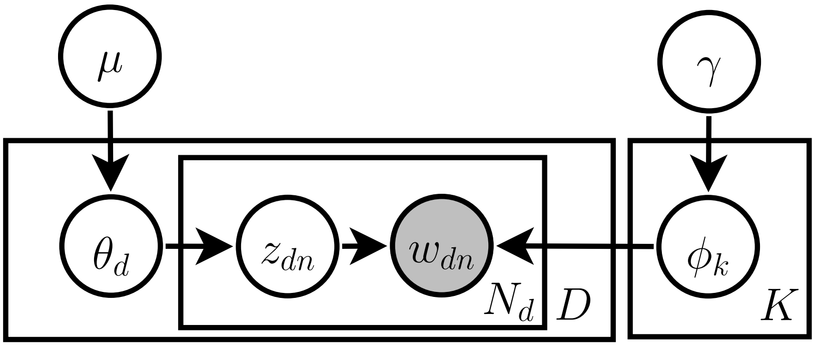


Figure 4.1: Graphical model of the latent Dirichlet allocation (LDA). The shaded node represents observed variable while the unshaded nodes represent latent variables.

[Hofmann, 1999]. The LDA can also be seen as a type of principal component analysis for discrete data [Buntine, 2002].

The LDA is an admixture model — each word in a document is assigned to a topic and hence a document is linked to multiple topics, rather than having only a topic per document. The Bayesian model of the LDA is given by the following generative process:

$$(\theta_d \mid \mu) \sim \text{Dirichlet}(\mu)\,, \qquad \text{for } d = 1, \dots, D\,, \tag{4.1}$$

$$(\phi_k \mid \gamma) \sim \text{Dirichlet}(\gamma)\,, \qquad \text{for } k = 1, \dots, K\,, \tag{4.2}$$

$$(z_{dn} \mid \theta_d) \sim \text{Discrete}(\theta_d)\,, \tag{4.3}$$

$$(w_{dn} \mid z_{dn}, \phi) \sim \text{Discrete}(\phi_{z_{dn}})\,, \qquad \text{for } n = 1, \dots, N_d\,. \tag{4.4}$$

In the above, μ and γ are the parameters for priors θ_d and ϕ_k respectively,[7] while z_{dn} is a topic index (*i.e.*, a label for a particular topic, usually numbered) and w_{dn} is a word associated with document d and position n (the n-th word in the text sequence); k is used to index the topics (out of K seen topics during sampling). Figure 4.1 shows the graphical model for the LDA.

Under this model, our aim is to infer the latent variables θ and ϕ, which are known as *document–topic distribution* and *topic–word distribution* respectively. Inference can be performed easily *via* the collapsed Gibbs sampling, in which the conjugacy between the distributions in the model allows a marginal posterior distribution to be derived. The Gibbs sampling is performed on the latent variable z, with the priors θ and ϕ being integrated out, even though the main interest is on them. This is because θ and ϕ can be constructed rather easily once we have the sample z.

Due to the simplicity of the LDA and its ease of implementation, it has been used widely in a variety of applications. It is also easily extended into a more compli-

[7] Here, the notation Dirichlet(a) represents the symmetric Dirichlet distribution with parameter $a = (a, a, \dots, a)$.

cated model for complex problems. A straightforward extension of the LDA is the hierarchical Dirichlet process LDA (HDP-LDA) [Teh *et al.*, 2006], which is a Bayesian nonparametric generalisation of the LDA. One advantage of nonparametric modelling is that it allows us to overcome a limitation of the LDA, for which the number of topics is a fix constant. The HDP-LDA relaxes this constraint and is able to learn the number of topics directly from the data. We note that the HDP-LDA is a special case of the hierarchical Pitman-Yor process LDA. We will revisit this in Chapter 5.

4.2 Topic Modelling with Metadata

Another extension to the LDA makes use of metadata, or auxiliary information that accompanying a document, for instance, *tweets* (short document from Twitter) contain additional information like authors, tags, and hyperlinks. This information is often discarded and ignored in a vanilla topic model such as the LDA.

In the context of microblog, such as *tweets*, each document is limited to a certain size[8] and usually contains informal languages (deliberate misspellings, acronyms, and abbreviations). Previous finding [Zhao *et al.*, 2011] suggests that the LDA does not work as well as other models that use metadata, as topics obtained from the LDA are mostly incoherent and not interpretable. A natural treatment to this is by aggregating these microblog documents together based on the authors to form documents that are larger [Weng *et al.*, 2010; Hong and Davison, 2010].

Instead of employing an *ad-hoc* approach in improving the LDA, a better solution would be to design a topic model that is more suitable in modelling the documents. Topic models that make use of metadata include author-topic model [Rosen-Zvi *et al.*, 2004], tag-topic model [Tsai, 2011], relational topic model [Chang and Blei, 2010], supervised LDA [Mcauliffe and Blei, 2008], Twitter-LDA [Zhao and Jiang, 2011], Topic-Link LDA [Liu *et al.*, 2009], and others. These models are able to make additional inference on documents, such as obtaining the word distributions correspond to certain authors or tags.

4.2.1 Author-topic Model

The author-topic model proposed by Rosen-Zvi *et al.* [2004] makes use of authorship information to improve topic modelling, it is a combination of both the LDA and the author model. The author model is analogous to a topic model, but with words generated from author-word distributions rather than topic–word distributions. The author model is not an admixture model like the LDA.

In the author-topic model, a new latent variable x is introduced, which serves to assign a word to an author. Hence, each word under this model is assigned a topic

[8] A tweet was limited to be 140 characters or less.

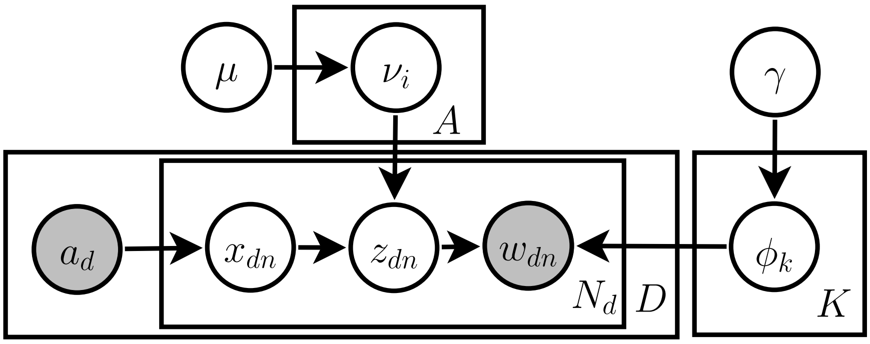


Figure 4.2: Graphical model for the author-topic model (ATM). As before, the shaded node represents observed variable while the unshaded represent latent variables.

and an author. The generative model for the author-topic model can be summarised as follows:

$$
\begin{aligned}
(\nu_i \,|\, \mu) &\sim \text{Dirichlet}(\mu)\,, && \text{for } i = 1, \ldots, A\,, && (4.5)\\
(\phi_k \,|\, \gamma) &\sim \text{Dirichlet}(\gamma)\,, && \text{for } k = 1, \ldots, K\,, && (4.6)\\
(x_{dn} \,|\, a_d) &\sim \text{Uniform}(a_d)\,, && \text{for } d = 1, \ldots, D\,, \quad n = 1, \ldots, N_d\,, && (4.7)\\
(z_{dn} \,|\, x_{dn}, \nu) &\sim \text{Discrete}(\nu_{x_{dn}})\,, && && (4.8)\\
(w_{dn} \,|\, z_{dn}, \phi) &\sim \text{Discrete}(\phi_{z_{dn}})\,, && && (4.9)
\end{aligned}
$$

Here, ν_i is the author–topic distribution for author i, which is used in generating the latent topic z_{dn} given the latent author x_{dn} , who is assumed to have written the word n in document d. Figure 4.2 shows the graphical model of author-topic model.

Note that the latent author x_{dn} is generated uniformly[9] from a_d , the list of authors in document d. This means that each word in document d is assumed to be contributed randomly by one of the authors. However, this assumption is not realistic, since a document is often written by the first author, and then adjusted by the others. In addition, the assumption fails to recognise the dependency of the words in term of authorship, that is, consecutive words tend to be penned down by the same person. A relaxation of this assumption would be to induce asymmetry in authorship and/or to assign authorship given the structure of the documents.

4.2.2 Tag-topic Model

The tag-topic model [Tsai, 2011] is essentially the same as author-topic model, except that the authorship information is replaced by tags. The model is arguably better than author-topic model as tags are more closely related to topics than authors, in

[9] We denote Uniform(b) to be a discrete uniform distribution for which the random outcome is one of the value from b chosen randomly with probability $1/|b|$.

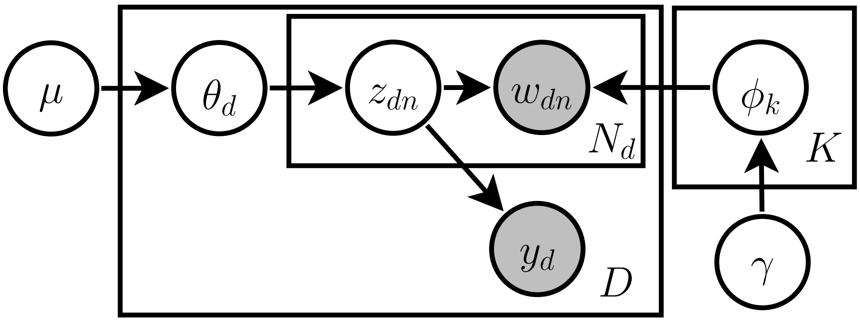


Figure 4.3: Graphical model for the supervised LDA. The shaded node represents observed variable while the unshaded nodes represent latent variables.

fact, a tag can be seen as a topic. The graphical model of the tag-topic model is omitted as it is almost identical to that of author-topic model. There are several topic model variants that also utilise tag information, this includes the TagLDA [Zhu *et al.*, 2006] and the Tag-LDA (with hyphen) [Si and Sun, 2009].

4.2.3 Supervised LDA

In contrast to author-topic model and tag-topic model where the metadata are used in generating the words in a document, supervised LDA deals with metadata that is generated from the model, like the generation of words. Supervised LDA works with any metadata that are relevant to a document. For example, movie ratings (score) for movie reviews (text). As such, supervised LDA can also be used to predict any quantity of interest (metadata) given the text data.

The graphical model for supervised LDA is given in Figure 4.3. Under this model, in addition to the usual generative process given by standard LDA (see Section 4.1), the observed variable y_d is generated by

$$(y_d \,|\, z_d, \beta, \delta) \sim GLM(\bar{z}_d, \beta, \delta) \,, \tag{4.10}$$

where $GLM(x, \beta, \delta)$ denotes the generalised linear model [McCullagh, 1984] with covariates x, regression parameters β and dispersion parameter δ. The probability density function for GLM is given by

$$p(y \,|\, x, \beta, \delta) = h(y, \delta) \exp\left(\frac{(x \cdot \beta)y - A(x \cdot \beta)}{\delta}\right) . \tag{4.11}$$

Here, the functions $h(y, \delta)$ and $A(x \cdot \beta))$ are known as *link function* and *log-normaliser*. For more details, refer to Mcauliffe and Blei [2008], and McCullagh [1984].

Note that supervised LDA represents each z_{dn} as a vector, for which exactly one value in the vector is 1 (the rest being 0). The explanatory power to predict y_d is then given by the mean vector $\bar{z}_d$, which is the *topic proportion* for document *d*:

$$\bar{z}_d = \frac{\sum_{n=1}^{N_d} z_{dn}}{N_d} . \tag{4.12}$$

4.3 Other Topic Models

There are many topic models that are used, both in practice and in theory. This section gives an overview of some of the topic models in the literature.

There are several topic models that model the evolution of topics through time or chapters, namely the dynamic topic model [Blei and Lafferty, 2006], the infinite dynamic topic model [Ahmed and Xing, 2010], the continuous time dynamic topic model [Wang *et al.*, 2008], the topics over time model [Wang and McCallum, 2006], the online LDA [AlSumait *et al.*, 2008], and many more.

Topic models that take the document structure into account includes the adaptive topic model [Du *et al.*, 2012a], the segmented topic model [Du *et al.*, 2010], the sequential LDA [Du *et al.*, 2012b], the structured topic model [Du *et al.*, 2013], and the structural topic model [Wang *et al.*, 2011a]. In other cases, the words in a document are not modelled with a bag-of-word assumption (*i.e.*, the words are not independently generated). Word dependency is explored by the Hidden Topic Markov Model [Gruber *et al.*, 2007], the Bigram topic model [Wallach, 2006], and the topical n-grams [Wang *et al.*, 2007].

In addition to the topic models mentioned in Section 4.2, topic models that use metadata for specific purpose are the Topic-tag model and the User-topic tag model [Bundschus *et al.*, 2009] which are used in tagging system. The inheritance topic model [He *et al.*, 2009] uses citations to predict topic evolution; while the topic-sentiment mixture model [Mei *et al.*, 2007] and the joint sentiment/topic model [Lin and He, 2009] perform sentiment analysis. Other notable topic models include the correlated topic model [Blei and Lafferty, 2007] that treats the topics as not interchangeable and induce correlation between them, and the multi-grain topic model [Titov and McDonald, 2008b] that models both global and local topics in a corpus.

4.4 Summary

In this chapter, we discuss some notable topic models used in practice, which are Bayesian in nature. In particular, we describe the LDA, which is the most basic Bayesian topic model, and then we touch on the HDP-LDA. We also outline some topic models that incorporate auxiliary information in their modelling, which inspire our proposed topic models in the later chapters.

In the next chapter, we present a generic topic model that employs the hierarchical Pitman-Yor process (HPYP). We then discuss a general framework to implement the topic model, which will also be used for topic models that are more complicated.

Chapter 5

Model Design and Implementation

In this chapter, we will discuss the basic design of our nonparametric Bayesian topic models using hierarchical Pitman-Yor process (HPYP). In particular, we will introduce a simple topic model that will be extended later. We discuss the general inference algorithm for the topic model and *hyperparameters* optimisation. In addition, we will present an evaluation metric commonly used to evaluate topic models.

From this chapter onward, we depart from the literature review and instead focus on the research aspect of this dissertation. The findings of this chapter are currently in press [Lim *et al.*, 2016].

5.1 Introduction

Development of topic models is fundamentally motivated by their applications. Depending on the application, a specific topic model that is most suitable for the task should be designed and used. However, despite the ease of designing the model, the majority of time is spent on implementing, assessing, and redesigning it. This calls for a better designing routine that is more efficient, that is, spending less time in implementation and more time in model design and development.

We can achieve this by a higher level implementation of the algorithms for topic modelling. This has been made possible in other statistical domains by BUGS [Lunn *et al.*, 2000] or JAGS [Plummer, 2003], albeit they only work with standard probability distributions. Theoretically, BUGS and JAGS will work on LDA; however, in practice, running Gibbs sampling for LDA with BUGS and JAGS is horrendously slow, this is because their Gibbs samplers are uncollapsed and not optimised. Furthermore, BUGS and JAGS cannot be used in a model with stochastic processes, like the Gaussian process (GP) and the Dirichlet process (DP).

Besides BUGS and JAGS, there are other frameworks that are designed for performing inference for general model. Examples include the Adaptor Grammars (AG) [Johnson *et al.*, 2007], Infer.NET [Minka *et al.*, 2014] and the Hierarchical Bayes Compiler (HBC) [Daumé III, 2007]. The AG are built specifically for modelling natural language and can be used to learn grammars (separating suffix from words), collocations (multi-words), and others. The AG is also shown to perform topic modelling, which acts as a variation of LDA. The AG framework is very useful in modelling hierarchical structure that has a tree structure.

Infer.NET is a framework for running Bayesian inference, it is aimed to solve many different kinds of machine learning problems using a variety of methods. One limitation of Infer.NET is that it does not deal with nonparametric models such as the DP, and hence is of little use to Bayesian nonparametric methods. Similar to Infer.NET, the HBC is designed to allow quick implementation of hierarchical models. Although HBC seems very promising, the project appears to be abandoned. Also, both Infer.NET and HBC failed to cover an important aspect of Bayesian modelling, that is, the sampling of hyperparameters (estimating the parameters of the priors).

Creating a program like BUGS (Bayesian inference using Gibbs sampling) or JAGS (just another Gibbs sampler) is a daunting task; it involves software engineering and good programming insight. In the following, we present a framework that allows us to implement HPYP topic models efficiently, achieved by modularising the PYP in the topic model. This framework allows us to test variants of our proposed topic models without significant reimplementation, which saves us precious time and effort in the implementation phase. In the next section, we first describe a general hierarchical PYP topic model, before discussing the framework.

5.2 Hierarchical Pitman-Yor Process Topic Model

Latent Dirichlet Allocation (LDA) [Blei *et al.*, 2003] is the simplest Bayesian topic model used in modelling text, which also allows easy learning of the model. Teh and Jordan [2010] proposed the HDP-LDA, which utilises the DP as a nonparametric prior which allows a non-symmetric, arbitrary dimensional topic prior to be used. Furthermore, one can replace the Dirichlet prior on the word vectors with the Pitman-Yor Process (PYP) [Teh, 2006b], which models the power-law of word frequency distributions in natural language [Goldwater *et al.*, 2011], yielding significant improvement [Sato and Nakagawa, 2010].

In this section, we introduce a generic topic model named the HPYP topic model. The HPYP topic model is a simple network of PYP nodes since all distributions on the probability vectors are modelled by the PYP. For simplicity, we assume a topic model with three PYP layers, although in practice there is no limit to the number of PYP layers.

We present the graphical model of our generic topic model in Figure 5.1. At the root level, we have μ and γ which are distributed according to a PYP:

$$\mu \sim \mathrm{PYP}(\alpha^{\mu}, \beta^{\mu}, H^{\mu})\,, \tag{5.1}$$

$$\gamma \sim \mathrm{PYP}(\alpha^{\gamma}, \beta^{\gamma}, H^{\gamma})\,. \tag{5.2}$$

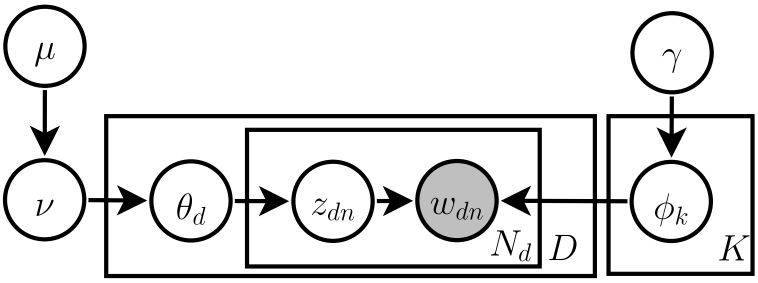


Figure 5.1: Graphical model of the HPYP topic model. It is an extension to LDA by allowing the probability vectors to be modelled by PYPs instead of the Dirichlet distributions. Left side of the graphical model (consists of μ, ν and θ) is referred as topic side, while the right hand side (with γ and ϕ) is called the vocabulary side. The word node denoted by w_{dn} is observed. Notations are defined in Table 5.1.

The variable μ is the root node for the *topics* in a topic model while γ is the root node for the *words*. To allow arbitrary number of topics to be learned, we let the base distribution for μ, H^μ, to be a continuous distribution or a discrete distribution with infinite samples. Note that the samples itself are ignored since we can relabel them to anything we like, thus the base distribution is of no significance. In this case, μ is also GEM distributed.

We usually choose a discrete uniform distribution for γ based on the word vocabulary size of the text corpus. This decision is technical in nature, as we are able to assign a tiny probability to words not observed in the training set, which eases the evaluation process. Thus $H^\gamma = \{\cdots, \frac{1}{|\mathcal{V}|}, \cdots\}$ where $|\mathcal{V}|$ is the set of all word vocabulary of the text corpus.

We now consider the topic side of the HPYP topic model. Here we have ν, which is the child node of μ. It follows a PYP given ν, which acts as its base distribution:

$$\nu \sim \mathrm{PYP}(\alpha^\nu, \beta^\nu, \mu)\,. \tag{5.3}$$

For each document d in a text corpus of size D, we have a document–topic distribution θ_d , which is a topic distribution specific to a document. Each of them tells us about the topic composition of a document.

$$\theta_d \sim \mathrm{PYP}(\alpha^{\theta_d}, \beta^{\theta_d}, \nu)\,, \qquad \text{for } d = 1, \ldots, D\,. \tag{5.4}$$

While for the vocabulary side, for each topic k learned by the model, we have a topic–word distribution ϕ_k which tells us about the words associated with each topic. The topic–word distribution ϕ_k is PYP distributed given the parent node γ:

$$\phi_k \sim \mathrm{PYP}(\alpha^{\phi_k}, \beta^{\phi_k}, \gamma)\,, \qquad \text{for } k = 1, \ldots, K\,. \tag{5.5}$$

Here, K is the number of topics in the topic model.

Table 5.1: List of variables for the HPYP topic model used in this chapter.

Variable	Name	Description
z_{dn}	Topic	Categorical/topical label for word w_{dn} .
w_{dn}	Word	Observed word or phrase at position n in document d.
ϕ_k	Topic–word distribution	Probability distribution in generating words for topic k.
θ_d	Document–topic distribution	Probability distribution in generating topics for document d.
γ	Global word distribution	Word prior for ϕ_k .
ν	Global topic distribution	Topic prior for θ_d .
μ	Global topic distribution	Topic prior for ν.
$\alpha^{\mathcal{N}}$	Discount	Discount parameter for PYP $\mathcal{N}$.
$\beta^{\mathcal{N}}$	Concentration	Concentration parameter for PYP $\mathcal{N}$.
$H^{\mathcal{N}}$	Base distribution	Base distribution for PYP $\mathcal{N}$.

For every word w_{dn} in a document d which is indexed by n (from 1 to N_d , the number of words in document d), we have a latent topic z_{dn} (also known as topic assignment) which indicates the topic the word represents. z_{dn} and w_{dn} are categorical variables generated from θ_d and ϕ_k respectively:

$$z_{dn} \,|\, \theta_d \sim \text{Discrete}(\theta_d)\,, \tag{5.6}$$

$$w_{dn} \,|\, z_{dn}, \phi \sim \text{Discrete}(\phi_{z_d})\,, \qquad \text{for } \; n = 1, \ldots, N_d\,. \tag{5.7}$$

The above α and β are the discount and concentration parameters of the PYPs (see Section 3.3.2), note that they are called the *hyperparameters* in the model. We present a list of variables used in this chapter in Table 5.1.

5.3 Model Representation and Posterior Likelihood

In a Bayesian setting, posterior inference for a topic model requires us to analyse the posterior distribution of the model variables given the observed data. For instance, the joint posterior distribution for the HPYP topic model is

$$p(\mu, \nu, \gamma, \theta, \phi, \mathbf{Z} \,|\, \mathbf{W}, \Xi)\,. \tag{5.8}$$

Here, we use bold face capital letters to represent the set of all relevant variables, as discussed in Section 1.4. For instance, **W** captures all words in the corpus. Additionally, we denote Ξ as the set of all hyperparameters and constants in the model.

Note that deriving the posterior distribution analytically is almost impossible due to its complex nature. This leaves us with approximate Bayesian inference techniques as discussed in Section 2.2. However, even with the above techniques, performing posterior inference with the posterior distribution is difficult due to the coupling of the probability vectors from the PYPs.

The key to an efficient inference procedure with the PYPs is to marginalise out the PYPs in the model and record various associated counts instead, which yields a collapsed sampler. To achieve this, we adopt a Chinese Restaurant Process (CRP) metaphor [Teh and Jordan, 2010; Blei *et al.*, 2010] to represent the variables in the topic model. With this metaphor, all data in the model (*e.g.*, topics and words) are the *customers*; while the PYP nodes are the *restaurants* the customers visit. In each restaurant, each customer is to be seated at only one *table*, though each table can have any number of customers. Each table in a restaurant serves a *dish*, the dish corresponds to the categorical label a data point may have (*e.g.*, the topic label or word). Note that there can be more than one table serving the same dish. In a HPYP topic model, the tables in a restaurant $\mathcal{N}$ are treated as the customers for the parent restaurant $\mathcal{P}$ (in the graphical model, $\mathcal{P}$ points to $\mathcal{N}$), and they share the same dish. This means that the data is passed up recursively until the root node. For illustration, we present a simple example in Figure 5.2, showing the seating arrangement of the customers from two restaurants.

Naïvely recording the seating arrangement (table and dish) of each customer brings about computational inefficiency during inference. Instead, we adopt the table multiplicity (or table counts) representation of Chen *et al.* [2011] which requires no dynamic memory, thus consuming only a factor of memory at no loss of inference efficiency. Under this representation, we store only the customer counts and table counts associated with each restaurant. The customer count $c_k^{\mathcal{N}}$ denotes the number of customers who are having dish k in restaurant $\mathcal{N}$. The corresponding symbol without subscript, $c^{\mathcal{N}}$, denotes the collection of customer counts in restaurant $\mathcal{N}$, that is, $c^{\mathcal{N}} = (\cdots, c_k^{\mathcal{N}}, \cdots)$. The total number of customers in a restaurant $\mathcal{N}$ is denoted by the capitalised symbol instead, $C^{\mathcal{N}} = \sum_k c_k^{\mathcal{N}}$. Similar to the customer count, the table count $t_k^{\mathcal{N}}$ denotes the number of non-empty tables serving dish k in restaurant $\mathcal{N}$. The corresponding $t^{\mathcal{N}}$ and $T^{\mathcal{N}}$ are defined similarly. For instance, from the example in Figure 5.2, we have $c_{\text{sun}}^2 = 9$ and $t_{\text{sun}}^2 = 3$, the corresponding illustration of the table multiplicity representation is presented in Figure 5.3. We refer the readers to Chen *et al.* [2011] for a detailed derivation of the posterior likelihood of a restaurant.

For the posterior likelihood of the HPYP topic model, we marginalise out the probability vector associated with the PYPs and represent them with the customer

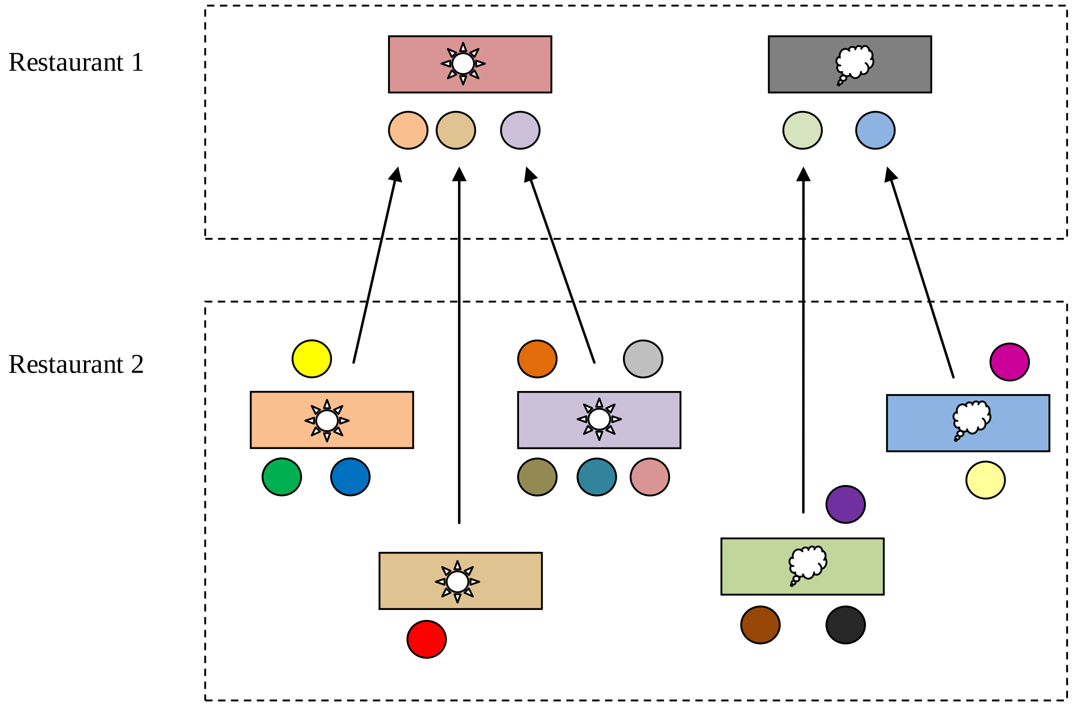


Figure 5.2: An illustration of the Chinese restaurant process representation. The customers are represented by the circles while the tables are represented by the rectangles. The dishes are the symbols in the middle of the rectangles, here they are denoted by the sunny symbol and the cloudy symbol. In this illustration, we know the number of customers corresponds to each table. For example, the green table is occupied by three customers. We note that there is no limit to the number of customers who can sit at a table. Also, since Restaurant 1 is the parent of Restaurant 2, the tables in Restaurant 2 are treated as the customers for Restaurant 1.

counts and table counts, following Chen *et al.* [2011, Theorem 1]. We present the modularised version of the full posterior of the HPYP topic model, which allows the posterior to be computed very quickly. The full posterior consists of the modularised likelihood associated with each PYP in the model, defined as

$$f(\mathcal{N}) = \frac{(\beta^{\mathcal{N}}|\alpha^{\mathcal{N}})_{T^{\mathcal{N}}}}{(\beta^{\mathcal{N}})_{C^{\mathcal{N}}}} \prod_{k=1}^{K} S^{c_k^{\mathcal{N}}}_{t_k^{\mathcal{N}},\alpha^{\mathcal{N}}} \binom{c_k^{\mathcal{N}}}{t_k^{\mathcal{N}}}^{-1} , \qquad \text{for } \mathcal{N} \sim \mathrm{PYP}(\alpha^{\mathcal{N}}, \beta^{\mathcal{N}}, \mathcal{P}) . \tag{5.9}$$

As mentioned in Section 3.2.4, $S^{x}_{y,\alpha}$ are generalised Stirling numbers [Buntine and Hutter, 2012, Theorem 17]. Both $(x)_T$ and $(x|y)_T$ denote the Pochhammer symbols (rising factorials) [Oldham *et al.*, 2009, Chapter 18]:

$$(x)_T = x \cdot (x+1) \cdots \big(x + (T-1)\big) \ , \tag{5.10}$$

$$(x|y)_T = x \cdot (x+y) \cdots \big(x + (T-1)y\big) \ . \tag{5.11}$$

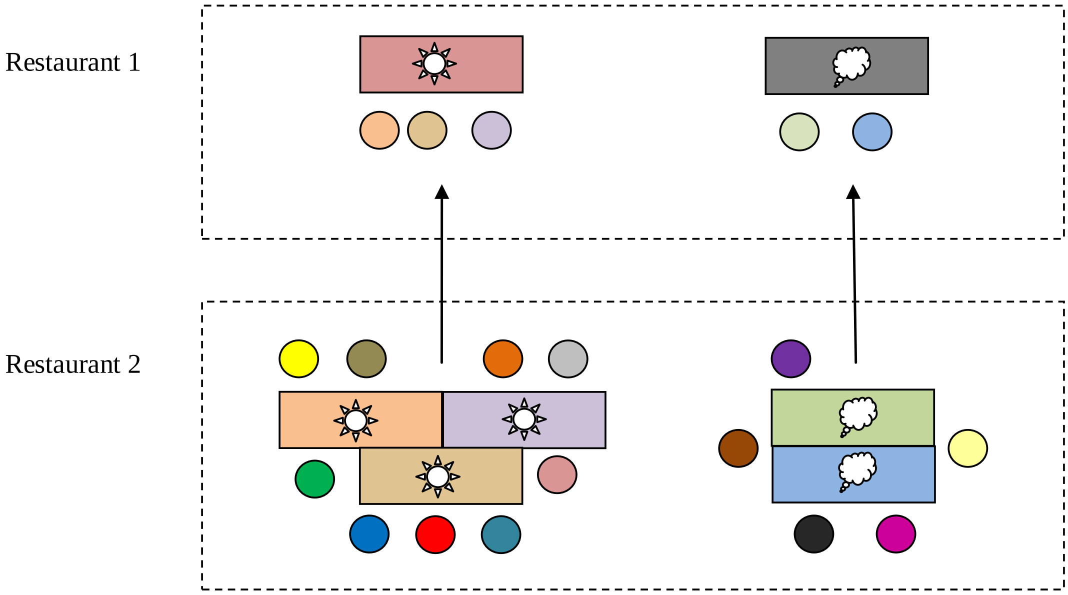


Figure 5.3: An illustration of the Chinese restaurant with the table counts representation. Here the setting is the same as Figure 5.2 but the seating arrangement of the customers are "forgotten" and only the table and customer counts are recorded. Thus, we can only know there are three sunny tables in Restaurant 2, and that there are nine customers sitting on those tables.

With the CRP representation, the full posterior of the HPYP topic model can now be written — in terms of $f(\cdot)$ given in Equation (5.9) — as

$$\begin{aligned} p(\mathbf{Z},\mathbf{T},\mathbf{C}\,|\,\mathbf{W},\Xi) &\propto p(\mathbf{Z},\mathbf{W},\mathbf{T},\mathbf{C}\,|\,\Xi) \\ &\propto f(\mu)f(\nu)\Bigg(\prod_{d=1}^{D} f(\theta_d)\Bigg)\Bigg(\prod_{k=1}^{K} f(\phi_k)\Bigg)f(\gamma)\Bigg(\prod_{v=1}^{|\mathcal{V}|}\bigg(\frac{1}{|\mathcal{V}|}\bigg)^{t_v^{\gamma}}\Bigg)\,. \end{aligned} \tag{5.12}$$

The last term in Equation (5.12) corresponds to the base distribution of γ, and v indexes each unique word in vocabulary set $\mathcal{V}$. Note that the topic assignments $\mathbf{Z}$ are implicitly captured by the customer counts:

$$c_k^{\theta_d} = \sum_{n=1}^{N_d} I(z_{dn} = k)\,, \tag{5.13}$$

where $I(\cdot)$ is the indicator function, which evaluates to 1 when the statement inside the function is true, and 0 otherwise. We would like to point out that even though the probability vectors of the PYPs are integrated out and not explicitly stored, they can easily be reconstructed. This is discussed in Section 5.4.4.

5.4 Posterior Inference for the HPYP Topic Model

We focus on the Markov chain Monte Carlo (MCMC) method for Bayesian inference on the HPYP topic model. The MCMC method on topic models follows these simple procedures — decrementing counts contributed by a word, sample a new topic for the word, and update the model by accepting or rejecting the proposed sample. Here, we describe the collapsed blocked Gibbs sampler for the HPYP topic model. Note the PYPs are marginalised out so we only deal with the counts. Recall that we will always accept the proposed sample in Gibbs sampling (see Section 2.2.1.2).

5.4.1 Decrementing the Counts Associated with a Word

The first step in a Gibbs sampler is to remove a word and corresponding latent topic, then decrementing the associated customer counts and table counts. To give an example from Figure 5.2, if we remove the red customer from Restaurant 2, we would decrement the customer count c^2_{sun} by 1. Additionally, we also decrement the table count t^2_{sun} by 1 because the red customer is the only customer on its table. This in turn decrements the customer count c^1_{sun} by 1. However, this requires us to keep track of the customers' seating arrangement which leads to increased memory requirements and poorer performance due to inadequate mixing [Chen *et al.*, 2011].

To overcome the above issue, we follow the concept of table indicator [Chen *et al.*, 2011] and introduce a new auxiliary Bernoulli indicator variable $u^{\mathcal{N}}_k$, which indicates whether removing the customer also removes the table by which the customer is seated. Note that our Bernoulli indicator is different to that of Chen *et al.* [2011] which indicates the restaurant a customer contributes to. The Bernoulli indicator is sampled as needed in the decrementing procedure and it is not stored, this means that we simply "forget" the seating arrangements and re-sample them later when needed, thus we do not need to store the seating arrangement. The Bernoulli indicator of a restaurant $\mathcal{N}$ depends solely on the customer counts and the table counts:

$$p(u^{\mathcal{N}}_k) = \begin{cases} t^{\mathcal{N}}_k / c^{\mathcal{N}}_k & \text{if } u^{\mathcal{N}}_k = 1 \\ 1 - t^{\mathcal{N}}_k / c^{\mathcal{N}}_k & \text{if } u^{\mathcal{N}}_k = 0 \ . \end{cases} \tag{5.14}$$

In the context of the HPYP topic model described in Section 5.2, we formally present how we decrement the counts associated with the word w_{dn} and latent topic z_{dn} from document d and position n. First, on the vocabulary side, we decrement the customer count $c^{\phi_{z_{dn}}}_{w_{dn}}$ associated with $\phi_{z_{dn}}$ by 1. Then sample a Bernoulli indicator $u^{\phi_{z_{dn}}}_{w_{dn}}$ according to Equation (5.14). If $u^{\phi_{z_{dn}}}_{w_{dn}} = 1$, we decrement the table count $t^{\phi_{z_{dn}}}_{w_{dn}}$ and also the customer count $c^{\gamma}_{w_{dn}}$ by one. In this case, we would sample a Bernoulli indicator $u^{\gamma}_{w_{dn}}$ for γ, and decrement $t^{\gamma}_{w_{dn}}$ if $u^{\gamma}_{w_{dn}} = 1$. We do not decrement the respective customer count if the Bernoulli indicator is 0. Second, we would need to

Table 5.2: All possible proposals for the blocked Gibbs sampler for the variables associated with w_{dn}. To illustrate, one sample would be $z_{dn} = 1$, $t^{\mathcal{N}}_{z_{dn}}$ does not increment (stays the same), and $c^{\mathcal{N}}_{z_{dn}}$ increments by 1, for all $\mathcal{N}$ in $\{\mu, \nu, \theta_d, \phi_{z_{dn}}, \gamma\}$. We note that the proposals can include states that are invalid, but this is not an issue since those states have zero posterior probability and thus will not be sampled.

Variable	Possibilities
z_{dn}	$1, \ldots, K$
$t^{\mathcal{N}}_{z_{dn}}$	$t^{\mathcal{N}}_{z_{dn}},\ t^{\mathcal{N}}_{z_{dn}} + 1$
$c^{\mathcal{N}}_{z_{dn}}$	$c^{\mathcal{N}}_{z_{dn}},\ c^{\mathcal{N}}_{z_{dn}} + 1$

decrement the counts associated with the latent topic z_{dn}. The procedure is similar, we decrement $c^{\theta_d}_{z_{dn}}$ by 1 and sample the Bernoulli indicator $u^{\theta_d}_{z_{dn}}$. Note that whenever we decrement a customer count, we sample the corresponding Bernoulli indicator. We repeat this procedure recursively until the Bernoulli indicator is 0 or until the procedure hits the root node.

5.4.2 Sampling a New Topic for a Word

After decrementing the variables associated with a word w_{dn}, we use a *blocked* Gibbs sampler to sample a new topic z_{dn} for the word and the corresponding customer counts and table counts. The conditional posterior used in sampling can be computed quickly when the full posterior is represented in a modularised form. To illustrate, the conditional posterior for z_{dn} and its associated customer counts and table counts is

$$p(z_{dn}, \mathbf{T}, \mathbf{C} \,|\, \mathbf{Z}^{-dn}, \mathbf{W}, \mathbf{T}^{-dn}, \mathbf{C}^{-dn}, \Xi) = \frac{p(\mathbf{Z}, \mathbf{T}, \mathbf{C} \,|\, \mathbf{W}, \Xi)}{p(\mathbf{Z}^{-dn}, \mathbf{T}^{-dn}, \mathbf{C}^{-dn} \,|\, \mathbf{W}, \Xi)}\,, \tag{5.15}$$

which is further broken down by substituting the posterior likelihood defined in Equation (5.12), giving the following ratios of the modularised likelihoods:

$$\frac{f(\mu)}{f(\mu^{-dn})} \frac{f(\nu)}{f(\nu^{-dn})} \frac{f(\theta_d)}{f(\theta_d^{-dn})} \frac{f(\phi_{z_{dn}})}{f(\phi_{z_{dn}}^{-dn})} \frac{f(\gamma)}{f(\gamma^{-dn})} \left(\frac{1}{|\mathcal{V}|}\right)^{t^{\gamma}_{w_{dn}} - \left(t^{\gamma}_{w_{dn}}\right)^{-dn}} . \tag{5.16}$$

The superscript $\square^{-dn}$ indicates that the variables associated with the word w_{dn} are removed from the respective sets, that is, the customer counts and table counts are after the decrementing procedure. Since we are only sample the topic assignment z_{dn} associated with one word, the customer counts and table counts can only increment by at most 1, see Table 5.2 for a list of all possible proposals. This allows the ratios of the modularised likelihoods, which consists of ratios of Pochhammer symbol and ratio of Stirling numbers,

$$\frac{f(\mathcal{N})}{f(\mathcal{N}^{-dn})} = \frac{(\beta^{\mathcal{N}})_{(C^{\mathcal{N}})^{-dn}}}{(\beta^{\mathcal{N}})_{C^{\mathcal{N}}}} \frac{(\beta^{\mathcal{N}}|\alpha^{\mathcal{N}})_{T^{\mathcal{N}}}}{(\beta^{\mathcal{N}}|\alpha^{\mathcal{N}})_{(T^{\mathcal{N}})^{-dn}}} \prod_{k=1}^{K} \frac{S^{c_k^{\mathcal{N}}}_{t_k^{\mathcal{N}},\, \alpha^{\mathcal{N}}}}{S^{(c_k^{\mathcal{N}})^{-dn}}_{(t_k^{\mathcal{N}})^{-dn},\, \alpha^{\mathcal{N}}}} \,, \tag{5.17}$$

to simplify further. For instance, the ratios of Pochhammer symbols can be reduced to constants, as follows:

$$\frac{(x)_{T+1}}{(x)_T} = x + T \,, \qquad\qquad \frac{(x|y)_{T+1}}{(x|y)_T} = x + yT \,. \tag{5.18}$$

While the ratio of Stirling numbers, such as $S^{y+1}_{x+1,\alpha} / S^{y}_{x,\alpha}$, can be computed quickly via caching [Buntine and Hutter, 2012]. We present a discussion on the Stirling numbers caching in Section 5.6.

With the conditional posterior defined, we proceed to the sampling process. Our first step involves finding all possible changes to the topic z_{dn}, customer counts, and the table counts (hereafter known as '*state*') associated with adding the removed word w_{dn} back into the topic model. Since only one word is added into the model, the customer counts and the table counts can only increase by at most 1, which limits the possible states to a reasonably small number. Furthermore, the customer counts of a parent node will only be incremented when the table counts of its child node increases. Note that it is possible for the added customer to generate a new dish (topic) for the model. This requires the customer to increment a new table count of the root node μ by one, the new table is associated with the new topic.

Next, we compute the conditional posterior (Equation (5.15)) for all possible states. As discussed, the conditional posterior (up to a proportional constant) can be computed quickly by breaking down the posterior and calculating the relevant parts. We then normalise them to sample one of the states to be the proposed next state. Note that the proposed state will always be accepted.

Finally, given the proposal, we update the HPYP model by incrementing the relevant customer counts and table counts. The technical details on the sampling process is presented in Section 5.6 for interested readers.

5.4.3 Optimising the Hyperparameters

Choosing the right hyperparameters for the priors is important for topic models. Wallach *et al.* [2009a] show that an optimised hyperparameter increases the robustness of the topic models and improves their model fitting. The hyperparameters of the HPYP topic models are the discount parameters and concentration parameters of the PYPs. Here, we outline the procedure to optimise the concentration parameters, but leave the discount parameters fixed due to the coupling of the discount parameters and the Stirling numbers cache.

The concentration parameters β of all the PYPs are optimised using an auxiliary variable sampler [Teh, 2006a]. Being Bayesian, we assume the concentration parameter $\beta^{\mathcal{N}}$ of a PYP node $\mathcal{N}$ has the following *hyperprior* distribution:

$$\beta^{\mathcal{N}} \sim \text{Gamma}(\tau_0, \tau_1)\,, \qquad\qquad \text{for } \mathcal{N} \sim \text{PYP}\big(\alpha^{\mathcal{N}}, \beta^{\mathcal{N}}, \mathcal{P}\big)\,, \tag{5.19}$$

where τ_0 is the *shape* parameter and τ_1 is the *rate* parameter. With the gamma prior, we have restricted the value of $\beta^{\mathcal{N}}$ to be strictly positive $(\beta^{\mathcal{N}} > 0)$ instead of greater than the negative of the discount parameter $(\beta^{\mathcal{N}} > -\alpha^{\mathcal{N}})$. The reason we choose the gamma prior is because it is a conjugate prior which gives a gamma posterior for $\beta^{\mathcal{N}}$. Additionally, this gives us a simple algorithm to optimise $\beta^{\mathcal{N}}$.

Before we are able to optimise $\beta^{\mathcal{N}}$, we first sample the auxiliary variables ω and ζ_i given the current value of $\alpha^{\mathcal{N}}$ and $\beta^{\mathcal{N}}$, as follows:

$$\omega \,|\, \beta^{\mathcal{N}} \sim \text{Beta}\big(C^{\mathcal{N}}, \beta^{\mathcal{N}}\big)\,, \tag{5.20}$$

$$\zeta_i \,|\, \alpha^{\mathcal{N}}, \beta^{\mathcal{N}} \sim \text{Bernoulli}\left(\frac{\beta^{\mathcal{N}}}{\beta^{\mathcal{N}} + i\alpha^{\mathcal{N}}}\right)\,, \qquad \text{for } i = 0, 1, \ldots, T^{\mathcal{N}} - 1\;. \tag{5.21}$$

With these, we can then sample a new $\beta^{\mathcal{N}}$ from its conditional posterior

$$\beta^{\mathcal{N}} \,|\, \omega, \zeta \sim \text{Gamma}\left(\tau_0 + \sum_{i=0}^{T^{\mathcal{N}}-1} \zeta_i \;\;,\;\; \tau_1 - \log(1-\omega)\right)\;, \tag{5.22}$$

where 'log' refers to the natural logarithm when the base is not specified. Note that instead of sampling a single value for each $\beta^{\mathcal{N}}$, we could repeat the procedure multiple times to obtain a simulated average for $\beta^{\mathcal{N}}$. Alternatively, we could also simply take the mean of the conditional posterior as an estimate for $\beta^{\mathcal{N}}$, as follows:

$$\mathbb{E}[\beta^{\mathcal{N}} \,|\, \omega, \zeta] = \frac{\tau_0 + \sum_i \zeta_i}{\tau_1 - \log(1-\omega)}\;. \tag{5.23}$$

In the collapsed Gibbs sampler, hyperparameter sampling is performed once every few iterations to update the hyperparameters. We summarise the collapsed Gibbs sampler in Algorithm 5.1.

5.4.4 Estimating the Probability Vectors of the PYPs

Recall that the aim of topic modelling is to analyse the posterior of the model parameters, such as one in Equation (5.8). Although we have marginalised out the PYPs in the above Gibbs sampler, the PYPs can be reconstructed from the associated customer counts and table counts. Recovering the full posterior distribution of the PYPs

Algorithm 5.1 Collapsed Gibbs Sampler for the HPYP Topic Model

```
1. Initialise the HPYP topic model by assigning random topic to the latent topic
   $z_{dn}$ associated to each word $w_{dn}$. Then update all the relevant customer counts
   $\mathbf{C}$ and table counts $\mathbf{T}$ by using Equation (5.13) and setting the table counts to
   be about half of the customer counts.
2. For each word $w_{dn}$ in each document $d$, do the following:
   (a) Decrement the counts associated with $w_{dn}$ (see Section 5.4.1).
   (b) Blocked-sample a new topic for $z_{dn}$ and corresponding customer counts $\mathbf{C}$
       and table counts $\mathbf{T}$ (see Section 5.4.2).
   (c) Update (increment counts) the topic model based on the sample.
3. Update the hyperparameter $\beta^{\mathcal{N}}$ for each PYP nodes $\mathcal{N}$ (see Section 5.4.3).
4. Repeat Steps 2 – 3 until the model converges or when a fix number of iterations
   is reached.
```

is a complicated task. So, instead, we will analyse the PYPs *via* the expected value of their conditional marginal posterior distribution, or simply, their *posterior mean*,

$$\mathbb{E}[\mathcal{N} \mid \mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}, \Xi]\,, \qquad\qquad \text{for } \mathcal{N} \in \{\mu, \nu, \gamma, \theta_d, \phi_k\}\,. \tag{5.24}$$

The posterior mean of a PYP corresponds to the probability of sampling a new customer for the PYP. To illustrate, we consider the posterior of the topic distribution θ_d . We let $\tilde{z}_{dn}$ to be an unknown *future* latent topic in addition to the known $\mathbf{Z}$. With this, we can write the posterior mean of θ_{dk} as

$$\begin{aligned}\mathbb{E}[\theta_{dk} \mid \mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}, \Xi] &= \mathbb{E}[p(\tilde{z}_{dn} = k \mid \theta_d, \mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}, \Xi) \mid \mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}, \Xi] \\ &= \mathbb{E}[p(\tilde{z}_{dn} = k \mid \mathbf{Z}, \mathbf{T}, \mathbf{C}) \mid \mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}, \Xi]\,. \end{aligned} \tag{5.25}$$

by replacing θ_{dk} with the posterior predictive distribution of $\tilde{z}_{dn}$ and note that $\tilde{z}_{dn}$ can be sampled using the CRP, as follows:

$$p(\tilde{z}_{dn} = k \mid \mathbf{Z}, \mathbf{T}, \mathbf{C}) = \frac{(\alpha^{\theta_d} T^{\theta_d} + \beta^{\theta_d})\nu_k + c_k^{\theta_d} - \alpha^{\theta_d} T_k^{\theta_d}}{\beta^{\theta_d} + C^{\theta_d}}\,. \tag{5.26}$$

Thus, the posterior mean of θ_d is given as

$$\mathbb{E}[\theta_{dk} \mid \mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}, \Xi] = \frac{(\alpha^{\theta_d} T^{\theta_d} + \beta^{\theta_d})\mathbb{E}[\nu_k \mid \mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}, \Xi] + c_k^{\theta_d} - \alpha^{\theta_d} T_k^{\theta_d}}{\beta^{\theta_d} + C^{\theta_d}}\,, \tag{5.27}$$

which is written in term of the posterior mean of its parent PYP, ν. The posterior means of the other PYPs such as ν can be derived by taking a similar approach. This is achieved by introducing an additional variable that serves as a customer to the particular PYP (just like $\tilde{z}_{dn}$). Generally, the posterior mean corresponds to a PYP $\mathcal{N}$ (with parent PYP $\mathcal{P}$) is as follows:

$$\mathbb{E}[\mathcal{N}_k \,|\, \mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}, \Xi] = \frac{(\alpha^{\mathcal{N}} T^{\mathcal{N}} + \beta^{\mathcal{N}})\mathbb{E}[\mathcal{P}_k \,|\, \mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}, \Xi] + c_k^{\mathcal{N}} - \alpha^{\mathcal{N}} T_k^{\mathcal{N}}}{\beta^{\mathcal{N}} + C^{\mathcal{N}}}, \tag{5.28}$$

By applying Equation (5.28) recursively, we obtain the posterior mean for all the PYPs in the model.

We note that the dimension of the topic distributions (μ, ν, θ) is $K + 1$, where K is the number of observed topics. This accounts for the generation of a new topic associated with the new customer, though the probability of generating a new topic is usually much smaller. In practice, we may instead ignore the extra dimension during the evaluation of a topic model since it does not provide useful interpretation. One way to do this is to simply discard the extra dimension of all the probability vectors after computing the posterior mean. Another approach would be to normalise the posterior mean of the root node μ after discarding the extra dimension, before computing the posterior mean of others PYPs. Note that for a considerably large corpus, the difference in the above approaches would be too small to notice.

5.5 Evaluations on Topic Models

Generally, there are two ways to evaluate a topic model. The first is to evaluate the topic model based on the task it performs, for instance, the ability to make predictions. The second approach is the statistical evaluation of the topic model on modelling the data, which is also known as the goodness-of-fit test. In this section, we will present some commonly used evaluation metrics (not exhaustive) that are applicable to all topic models, but we first discuss the procedure for estimating variables associated with the test set.

5.5.1 Predictive Inference on the Test Documents

Test documents, which are used for evaluations, are set aside during Gibbs sampling. As such, the document–topic distributions $\tilde{\theta}$ associated with the test documents are unknown and hence need to be estimated. Note we have used the symbol tilde ($\tilde{}$) to represent the variables from the test set. One estimate for $\tilde{\theta}$ is its posterior mean given the variables learned from the Gibbs sampler:

$$\hat{\tilde{\theta}}_d = \mathbb{E}[\tilde{\theta}_d \,|\, \mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}, \Xi]\,, \tag{5.29}$$

obtainable by applying Equation (5.28). Note that since the latent topics $\tilde{\mathbf{Z}}$ corresponding to the test set are not sampled, the customer counts and table counts associated with $\tilde{\theta}_d$ are 0, thus $\hat{\tilde{\theta}}_d$ is equal to $\hat{\nu}$, the posterior mean of ν. However, this is not a good estimate for the topic distribution of the test documents since they will be identical for all the test documents. To overcome this issue, we will instead use some of the words in the test documents to obtain a better estimate for $\tilde{\theta}$. This method is known as document completion [Wallach *et al.*, 2009b], as we use part of the text to estimate $\tilde{\theta}$, and use the rest for evaluation. The common split in practice is fifty-fifty, but it is also reasonable to use a smaller amount of words for estimation if the text is reasonably long.

Getting a better estimate for $\tilde{\theta}$ requires us to first sample some of the latent topics $\tilde{z}_{dn}$ in the test documents. The proper way to do this is by running an algorithm akin to the collapsed Gibbs sampler, but this would be excruciatingly slow due to the need to re-sample the customer counts and table counts for all the parent PYPs. Instead, we assume that the variables learned from the Gibbs sampler are fixed and sample the $\tilde{z}_{dn}$ from their conditional posterior sequentially, given the previous latent topics:

$$p(\tilde{z}_{dn} = k \,|\, \tilde{w}_{dn}, \tilde{\theta}_d, \phi, \tilde{z}_{d1}, \ldots, \tilde{z}_{d,n-1}) \propto \tilde{\theta}_{dk} \, \phi_{k w_{dn}} \,. \tag{5.30}$$

Whenever a latent topic $\tilde{z}_{dn}$ is sampled, we increment the customer count $c^{\tilde{\theta}_d}_{\tilde{z}_{dn}}$ for the test document. For simplicity, we set the table count $t^{\tilde{\theta}_d}_{\tilde{z}_{dn}}$ to be half the corresponding customer counts $c^{\tilde{\theta}_d}_{\tilde{z}_{dn}}$, this avoids the expensive operation of sampling the table counts. Additionally, $\tilde{\theta}_d$ is re-estimated using Equation (5.29) before sampling the next latent topic. We note that the estimated variables are unbiased.

The final $\tilde{\theta}_d$ becomes an estimate for the topic distribution of the test document d. The above procedure is repeated R times to give R samples of $\tilde{\theta}^{(r)}_d$, which are used to compute the following Monte Carlo estimate of $\tilde{\theta}_d$:

$$\hat{\tilde{\theta}}_d = \frac{1}{R} \sum_{r=1}^{R} \tilde{\theta}^{(r)}_d \,. \tag{5.31}$$

The Monte Carlo estimate can then be used for computing the evaluation metrics. Note that when estimating $\tilde{\theta}$, we have ignored the possibility of generating a new topic, that is, the latent topics $\tilde{z}$ are constrained to the existing topics, as previously discussed in Section 5.4.4.

5.5.2 Goodness-of-fit Test

There are multiple ways to perform a goodness-of-fit test on statistical models, such as calculating the mean square error in a regression model. Measures of goodness-of-fit usually involves computing the discrepancy of the observed values and the predicted values under the model. However, the observed variables in a topic model are the words in the corpus, which are not quantifiable since they are discrete labels. Thus evaluations on topic models are usually based on the model likelihoods instead.

A popular metric commonly used to evaluate the goodness-of-fit of a topic model is perplexity, which is negatively related to the likelihood of the observed words **W** given the model, this is defined as

$$\text{perplexity}(\mathbf{W} \mid \theta, \phi) = \exp\left(- \frac{\sum_{d=1}^{D} \sum_{n=1}^{N_d} \log p(w_{dn} \mid \theta_d, \phi)}{\sum_{d=1}^{D} N_d} \right), \tag{5.32}$$

where $p(w_{dn} \mid \theta_d, \phi)$ is the likelihood of sampling the word w_{dn} given the document–topic distribution θ_d and the topic–word distributions ϕ. Computing $p(w_{dn} \mid \theta_d, \phi)$ requires us to marginalise out z_{dn} from their joint distribution, as follows:

$$\begin{aligned} p(w_{dn} \mid \theta_d, \phi) &= \sum_k p(w_{dn}, z_{dn} = k \mid \theta_d, \phi) \\ &= \sum_k p(w_{dn} \mid z_{dn} = k, \phi_k)\, p(z_{dn} = k \mid \theta_d) \\ &= \sum_k \phi_{k w_{dn}} \theta_{dk} \ . \end{aligned} \tag{5.33}$$

Although perplexity can be computed on the whole corpus, in practice we compute the perplexity on test documents. This is to measure if the topic model generalises well to unseen data. A good topic model would be able to predict the words in the test set better, thereby assigning a higher probability $p(w_{dn} \mid \theta_d, \phi)$ in generating the words. Since perplexity is negatively related to the likelihood, a lower perplexity is better.

Note that the perplexity or simply the likelihood of a topic model can be used to assess the effectiveness of a training algorithm. By occasionally computing the perplexity (on the training set) during training, we can see whether if the model fitting is getting better, which is an important tool for topic model diagnostics.

5.5.3 Topic Similarity Analysis

In topic models, each individual word in the corpus is assigned a single topic label through the inference procedure. Ideally we would like all the words which are assigned to the same topic to be coherent and make sense as a topic. To evaluate the ability of a topic model to form coherent topics, we could manually analyse the topic–word distributions ϕ, by listing and inspecting the top words. The top n words of a topic–word distribution ϕ_k are obtained by querying words $v_1, \ldots, v_n$ for which the probability ϕ_{kv} are highest. However, this process can be tedious for topic models with large number of topics. Moreover, humans' judgement can be unreliable and inconsistent with one another.

A quantitative alternative would be to compute the pairwise Hellinger distance [Newman *et al.*, 2009] for each pair of topic–word distributions. The Hellinger distance is commonly used to measure the dissimilarity between two probability distributions, it is given as

$$\text{Hellinger}(\phi_i, \phi_j) = \frac{1}{\sqrt{2}} \left(\sum_{v=1}^{|\mathcal{V}|} \left(\sqrt{\phi_{iv}} - \sqrt{\phi_{jv}} \right)^2 \right)^{\frac{1}{2}} . \tag{5.34}$$

From Equation (5.34), we can see that the Hellinger distance for two similar probability vectors will be close to zero. We would like the topics to be as dissimilar as possible, thus the higher the Hellinger distance the better. The Hellinger distance is upper-bounded by one. To quickly visualise all K^2 Hellinger distances, we can display them as a heat chart, which gives a one-glance view of the similarity between the topics. Examples of the heat chart can be found in later chapters (*e.g.*, in Figure 6.4).

5.5.4 Document Clustering

Recall that topic models assign a topic to each word in a document, essentially performing a *soft clustering* [Erosheva and Fienberg, 2005] for the documents in which the membership is given by the document–topic distribution θ. To evaluate the clustering of the documents, we convert the soft clustering to hard clustering by choosing a topic that best represents the documents, hereafter called the *dominant topic*. The dominant topic of a document d corresponds to the topic that has the highest proportion in the topic distribution, that is,

$$\text{Dominant Topic}(\theta_d) = \arg\max_k \theta_{dk} . \tag{5.35}$$

Two commonly used evaluation measures for clustering are *purity* and *normalised mutual information* (NMI) [Manning *et al.*, 2008]. Purity is a simple clustering measure which can be interpreted as the proportion of documents correctly clustered, while NMI is an information theoretic measures used for clustering comparison. Here, we denote the ground truth classes as $\mathcal{S} = \{s_1, \ldots, s_J\}$ and the obtained clusters as $\mathcal{R} = \{r_1, \ldots, r_K\}$, where each s_i and r_i represents a collection (set) of documents. The purity and NMI can then be computed as

$$\text{purity}(\mathcal{S}, \mathcal{R}) = \frac{1}{D} \sum_{k=1}^{K} \max_j |r_k \cap s_j| , \qquad \text{NMI}(\mathcal{S}, \mathcal{R}) = \frac{2\,\text{MI}(\mathcal{S}; \mathcal{R})}{E(\mathcal{S}) + E(\mathcal{R})} , \tag{5.36}$$

where $\text{MI}(\mathcal{S}; \mathcal{R})$ denotes the mutual information between two sets and $E(\cdot)$ denotes the entropy. They are defined as follows:

$$\text{MI}(\mathcal{S}; \mathcal{R}) = \sum_{k=1}^{K} \sum_{j=1}^{J} \frac{|r_k \cap s_j|}{D} \log_2 D \frac{|r_k \cap s_j|}{|r_k||s_j|} , \quad E(\mathcal{R}) = - \sum_{k=1}^{K} \frac{|r_k|}{D} \log_2 \frac{|r_k|}{D} . \tag{5.37}$$

5.6 Implementation

To perform inference on a general topic model with a hierarchical PYP structure, we implemented a general topic modelling framework that modularise the PYP nodes. In this section, we briefly discuss the implementation of the general topic modelling framework, which is written in the *Java* programming language.

Our topic model framework consists of three parts, which are data preprocessing, model learning, and evaluation. Here we focus on model learning. We leave the data preprocessing discussion to the later chapters where they require different preprocessing techniques tailored to various data type. The implementation for evaluation is relatively straightforward and thus not discussed.

5.6.1 State

We first discuss the state of the model, which is a collection of variables used in the model. The state consists of all the PYP nodes of the model, the base distribution H^{γ}, and the topic assignment $\mathbf{Z}$. We briefly describe each part as follows.

5.6.1.1 PYP Node

Each PYP node $\mathcal{N}$ in the implementation framework stores the discount parameter $\alpha^{\mathcal{N}}$ and the concentration parameter $\beta^{\mathcal{N}}$, as well as the associated customer counts and table counts. Additionally, the PYP node also has a reference to its parent node (base distribution), allowing recursive operations to be performed easily.

The PYP node has routines (functions) to increment counts and decrement counts (and also sample the Bernoulli indicator used in decrementing counts). These procedures are recursive in that they call the respective routines of its parent node. In addition, the PYP node has a routine to sample a new concentration parameter $\beta^{\mathcal{N}}$, following the procedure in Section 5.4.3. Finally, the PYP node can also compute the modularised likelihood and the likelihood ratio according to Equation (5.9) and Equation (5.17), and estimate its posterior mean with Equation (5.28).

5.6.1.2 Base Distribution

Here, we describe the base distribution that is in the form of probability vector. An example of this is H^{γ}, which is a uniform vector. In our implementation, we treat the base distribution like a PYP node, we store the "customer counts" for the base distribution, which are just the table counts from its child node. Unlike PYP node, the base distribution does not have table counts. Note that although storing the customer counts and table counts for each PYP node may seem redundant, it is actually important for more complicated topic models in the later chapters.

The base distribution has similar routines to the PYP Node, with a major difference in the way of computing the likelihood. For instance, the modularised posterior likelihood for the base distribution H^{γ} corresponds to the last term in Equation (5.12).

5.6.1.3 Topic Assignments

The topics in the topic model are represented as a positive integer from 0 to $K-1$, where K is the number of topics. As such, the topic assignments in $\mathbf{Z}$ take values from 0 to $K-1$.

In our implementation, we store the topic assignments z_d for each document d as separate variables. Each of the z_d is a vector of z_{dn} and has a reference to its parent node, θ_d. The topic assignments z_d has a routine to initialise itself randomly, which also update the counts of the parent nodes recursively.

5.6.1.4 Customer Counts and Table Counts

The table counts t and the customer counts c for a PYP node $\mathcal{N}$ can be sparse, that is, most of the t_k and c_k are zeros.[10] For efficient storage of the table counts and the customer counts, we adopt the *OpenIntIntHashMap* from the *Colt* library,[11] which is a more efficient HashMap for integers.

Furthermore, we store the various sums of the table counts and the customer counts in a cache. This avoids the need to compute the sum repeatedly, thus speeds up the algorithm considerably.

5.6.2 Inference Procedure

Next we briefly discuss the procedure to perform one iteration of Gibbs sampling for each word w_{dn} in document d. Our first step is to decrement the counts associated with w_{dn} and z_{dn}, which is achieved by calling the decrement routine on the PYP nodes θ_d and $\phi_{z_{dn}}$. As discussed above, the PYP nodes recursively decrement the counts of their parent nodes based on the sampled Bernoulli indicators.

After decrementing the counts, we proceed with performing Gibbs sampling to sample a new z_{dn}^{new} and the associated counts. The Gibbs sampler first generate a list of all possible next states for the model, which are the combination of the next topic and whether or not to increment the customer counts and/or table counts. The conditional posteriors, as in Equation (5.15), are then computed for each of the possible state. Note that since the conditional posteriors can only be computed up to a proportional constant, we normalise them (so they sum up to 1) before sampling for one of the states. The sampled state then becomes the next state for the model.

We would like to point out that to prevent *underflow* and to maintain a better accuracy, the posterior likelihoods (and their ratios) are computed in log form. We also note in our implementation, we cache the posterior ratios of each PYP node to improve the algorithm speed, since they tend to occur many times in the calculation of all the possible next states.

[10] We ignore the superscript $\square^{\mathcal{N}}$ for better presentation.

[11] http://dst.lbl.gov/ACSSoftware/colt/ (last accessed 10 February 2013).

5.6.3 Stirling Numbers

Finally, we describe the implementation for the computation of the Stirling numbers. We first note that the ratio of Stirling numbers can be computed recursively [Buntine and Hutter, 2012] as follows:

$$U_{t,\alpha}^{c} = \frac{S_{t,\alpha}^{c+1}}{S_{t,\alpha}^{c}}\,, \qquad \text{for } c \geq t \geq 1\,, \tag{5.38}$$

$$V_{t,\alpha}^{c} = \frac{S_{t,\alpha}^{c}}{S_{t-1,\alpha}^{c}}\,, \qquad \text{for } c \geq t > 1\,. \tag{5.39}$$

The relationships of U and V are given as

$$U_{t,\alpha}^{c} = \frac{1}{V_{t,\alpha}^{c}} + c - t\alpha \qquad , \qquad \text{for } c \geq t > 1\,, \tag{5.40}$$

$$V_{t,\alpha}^{c} = \frac{1 + (c-1-t\alpha)V_{t,\alpha}^{c-1}}{U_{t-1,\alpha}^{c-1}}\,, \qquad \text{for } c \geq t > 1\,, \tag{5.41}$$

with the base case $U_{1,\alpha}^{c} = c - \alpha$ and noting that $V_{t,\alpha}^{c} = 0$ if $t > c$. In our implementation, we store only the values of V and compute U as needed. With these, we can then compute the Stirling numbers $S_{t,\alpha}^{c}$ recursively and store them in log form, since their values get exponentially larger. To conserve the memory, we store the Stirling numbers after every j-th increment in c, but storing them for every t, that is, we store $S_{t,\alpha}^{t+1}$, $S_{t,\alpha}^{t+1+j}$, $S_{t,\alpha}^{t+1+2j}$ and so on for every t. The unstored Stirling numbers are obtained *via* the following linear interpolation (in log form) from the nearest stored Stirling numbers:

$$\log S_{t,\alpha}^{t+1+nj+i} = \log S_{t,\alpha}^{t+1+nj} + \left(\log S_{t,\alpha}^{t+1+(n+1)j} - \log S_{t,\alpha}^{t+1+nj}\right) \times \frac{i}{j}\,. \tag{5.42}$$

Note that since the linear interpolation is only accurate when c is relatively larger than t (see Figure 5.4), we store the exact log Stirling numbers $S_{t,\alpha}^{c}$ in a separate table when $c - t < \rho$. In our implementation, we set $j = 20$ and $\rho = 40$, though they can easily be adjusted for different applications.

For large corpora of text the difference in c and t can be enormous, so it is not always feasible to store all the log Stirling numbers. In our implementation, we make use of the asymptotic expression of the Stirling numbers [Buntine and Hutter, 2012] when $c - t > \tau$, where τ is a threshold parameter. The asymptotic expression is

$$S_{t,\alpha}^{c} \approx \frac{1}{\Gamma(1-\alpha)} \frac{1}{\Gamma(t)\ \alpha^{t-1}} \frac{\Gamma(c)}{c^{\alpha}}\,, \qquad \text{for } \alpha > 0\,. \tag{5.43}$$

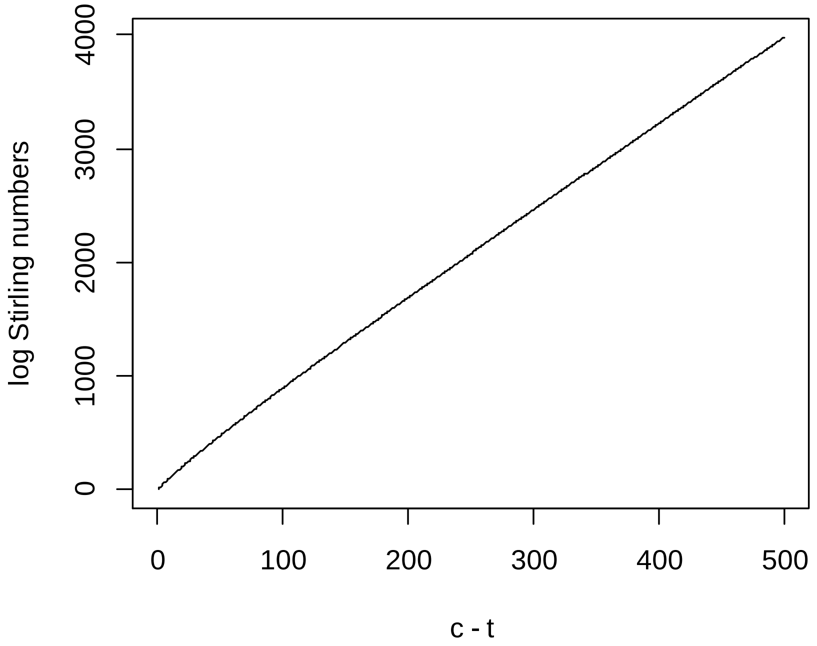


Figure 5.4: A plot of log Stirling numbers ($\log S^{c}_{t,\alpha}$) against the difference in counts ($c - t$) for $t = 1000$ and $\alpha = 0.5$. We can see that the log Stirling numbers are almost linear when the difference in c and t is large. This makes the linear interpolation a suitable method in estimating the unstored Stirling numbers.

In our experiments we find that the asymptotic estimates are more accurate as $c - t$ gets larger and as α approaches 1. Thus, we choose τ such that we do not compromise the accuracy of the Stirling numbers:

$$\tau = 4000 + \frac{1000}{\alpha} \,. \tag{5.44}$$

5.7 Summary

In this chapter, we introduce a general framework for modelling a hierarchical PYP topic model. We present a simple topic model that will be used as a skeleton for the later chapters. Using this topic model as an example, we walk through the model specification, model likelihood, and its inference procedures. Additionally, we discuss a few evaluation measures commonly used on topic models. These evaluations are applied to the topic models that we will introduce later. At the end of this chapter, we describe the implementation philosophy of our PYP topic models. Note that we only cover the non-trivial bits of our implementation in Section 5.6.

In the following chapters, we look into various topic models that are designed to model documents with auxiliary information. We will first start with a topic model that utilises hashtags and sentiment lexicons for opinion mining on tweets.

Chapter 6

Opinion Mining Using Hashtags, Emoticons and Sentiment Lexicon

Aspect-based opinion mining is widely applied to review data to aggregate or summarise opinions of a product. In this chapter, we introduce a topic model built upon the previously discussed principles for opinion mining and sentiment analysis. We name our model the *Twitter Opinion Topic Model* (TOTM), which as the name suggests, performs opinion mining on tweets. Tweets are often informal, unstructured and lacking labelled data such as categories and ratings, making it challenging for opinion mining. The TOTM leverages *hashtags*, *mentions*, emoticons and strong sentiment words that are present in tweets in its discovery process. It improves opinion prediction by modelling the target–opinion interaction directly, thus discovering target specific opinion words, neglected in existing approaches. Moreover, we propose a new formulation of incorporating sentiment prior information into a topic model, by utilising an existing public sentiment lexicon. This is novel in that it learns and updates with the data. This chapter is an extension of our published work in Lim and Buntine [2014b].

6.1 Introduction

When making a purchase decision, a key deciding factor can often be the reviews written by other consumers. These reviews are freely available online. However, one can rarely read all the reviews given their volume. This has led to various automated algorithms to mine the reviews, extracting a more digestible summary for a user. The task of analysing opinions from text data such as reviews is known as opinion mining or opinion extraction [Pang and Lee, 2008; Liu, 2012].

Among various approaches to opinion mining, *aspect-based opinion mining* has recently gained a lot of attention from the research community. Aspect-based opinion mining involves extracting the major aspects or facets from data for analysis. As an example, for a camera product, the aspects could be "picture quality", "portability", and others. Topic models are often used to determine the aspects through *soft clus-*

tering. They have also been successfully applied to review data crawled from review websites, such as *Epinions.com* and *TripAdvisor*. LDA-based[12] models are currently considered to be the state-of-the-art for aspect-based opinion mining [Moghaddam and Ester, 2012].

Besides reviews extracted from review websites, opinions from social media websites are also very useful, even though they are often overlooked as a source for reviews. Social media text is short and is regarded as "dirty", and hence less useful for more sophisticated language analysis [Zhao *et al.*, 2011]. The same problem also leads to degradation when applying NLP tools [Ritter *et al.*, 2011]. Despite these limitations, a large number of tweets containing opinions are generated every day and are very relevant for opinion mining. We argue that while tweets are generally unstructured, Twitter is a useful source of reviews since it provides a convenient platform for users to express their opinions. Twitter is also integrated to a person's social life, making it easier for users to express their opinions (on products, services, *etc.*) by tweeting instead of writing a review on review websites.

In this chapter, we demonstrate the usefulness of Twitter as a source for aspect-based target–opinion mining. We propose a novel LDA-based opinion model that is designed for tweets, which we name Twitter Opinion Topic Model (TOTM). TOTM models the target–opinion interaction directly, which significantly improves opinion prediction. For example, TOTM discovers that *'grilled'* is a positive opinion word for the target word *'sausage'*, but not for the other target words. We note that while there are no explicit ratings and scores on tweets, tweets often contain emoticons and strong sentiment words such as *'love'* and *'hate'*. TOTM exploits this fact and uses such information to compensate for the lack of explicit ratings. Additionally, hashtags are strong indicators of topics for tweets [Mehrotra *et al.*, 2013]. TOTM makes use of the hashtags and *mentions*[13] in tweets for tweet aggregation, which improves aspect clustering. Modelling with TOTM also allows us to acquire additional summaries on products, which are not obtainable with existing topic models.

Furthermore, we incorporate a sentiment lexicon as prior information into TOTM. We propose a novel formulation of how the sentiment lexicon affects the priors in TOTM. Our approach facilitates automatic learning of the lexicon strength based on the data; while current existing methods are *ad hoc* or ruled-based. Our formulation is shown to perform the best for sentiment classification. Additionally, we propose a different target–opinion extraction procedure that works better for tweets. We note that text preprocessing is important when dealing with tweets.

We apply TOTM on three tweet corpora, showing improved performance of TOTM in model fitting and sentiment analysis. In terms of application, we demonstrate the usefulness of TOTM in extracting the opinions on products from tweets.

[12] LDA is an acronym for Latent Dirichlet Allocation, as mentioned in previous chapters.

[13] Mentions are akin to user tagging, which are represented by the @ symbol. See https://support.twitter.com/articles/14023-what-are-replies-and-mentions (last accessed 11 June 2014) for details.

The tasks include opinion analysis on specific targets, brands opinion comparison, and extraction of constrastive opinions. As large volumes of tweets laden with opinions are generated daily, real-time aspect-based opinion analysis allows us to obtain first-hand opinions on new products, which might not be as readily available from review websites.

The rest of this chapter is structured as follows. Section 6.2 reviews recent work relevant to this chapter, and Section 6.3 provides a summary of our task and outlines the major contributions. In Section 6.4, we present and discuss the Interdependent LDA (ILDA) [Moghaddam and Ester, 2011], which will be used as a baseline for comparison. We introduce TOTM in Section 6.5 and the method of incorporating a lexicon in Section 6.6. In Section 6.7, we discuss the model likelihood and inference procedure of the TOTM, as well as propose a novel hyperparameter sampling procedure. We then describe the data used in this chapter in Section 6.8 and discuss how we preprocess the data. Next, we report on the experiments in Section 6.9 and perform model diagnostic in Section 6.10. Finally, we present a summary of this chapter in Section 6.11.

6.2 Related Work

LDA has been extended by many for sentiment analysis. Notable LDA-based topic models for sentiment analysis include the MaxEnt-LDA hybrid model [Zhao *et al.*, 2010], Joint Sentiment Topic model [Lin and He, 2009], Multi-grain LDA (MG-LDA) [Titov and McDonald, 2008b], Interdependent LDA (ILDA) [Moghaddam and Ester, 2011], Aspect and Sentiment Unification Model (ASUM) [Jo and Oh, 2011], and Multi-Aspect Sentiment model [Titov and McDonald, 2008a]. The Topic-Sentiment Mixture model [Mei *et al.*, 2007] performs sentiment analysis by using the multinomial distribution instead of the Dirichlet–multinomial distribution. These models perform aspect-based opinion analysis and they had been successfully applied to review data of different domains, such as electronic product, hotel and restaurant reviews. The task of summarising the reviews is also known as *opinion aggregation*.

To the best of our knowledge, there is no existing LDA-based opinion aggregation method other than ours that has been successfully applied to social media data such as tweets. Current opinion mining methods that are used on tweets tend to be *ad hoc* or rule-based. We suspect this is because tweets are generally regarded as too noisy for model-based methods to work, and also due to the fact that LDA works badly on short documents [Yan *et al.*, 2013]. Maynard *et al.* [2012] studied the challenges in developing an opinion mining tool for social media and they advocated the use of shallow techniques in linguistic processing of tweets. Notable non-LDA-based methods for opinion analysis include OPINE [Popescu and Etzioni, 2005], which uses relaxation labelling to classify sentiment, and Opinion Digger [Moghaddam and Ester, 2010], an aspect-based review miner using *k nearest neighbour*. Hu and Liu

[2004] performed rule-based target–opinion extraction from online product reviews, while Li *et al.* [2010] extracted opinions from reviews using *Conditional Random Fields*. On tweets, Pak and Paroubek [2010] performed opinion analysis using a Naive Bayes classifier; while Liu *et al.* [2013] performed sentiment classification with an adaptive co-training SVM. Go *et al.* [2009] and Davidov *et al.* [2010] made use of emoticons (smileys), which were found to provide improvement for sentiment classification on tweets. Since tweets are always short,[14] existing work [Go *et al.*, 2009; Pak and Paroubek, 2010; Davidov *et al.*, 2010; Liu *et al.*, 2013] tends to assume a single polarity for each tweet. In contrast, Jiang *et al.* [2011] performed target-dependent sentiment analysis, where the sentiments apply to a specific target.

Lexical information can be used to improve sentiment analysis. He [2012] used a sentiment lexicon to modify the priors of LDA for sentiment classification, though with a simple *ad hoc* approach. Li *et al.* [2009] incorporated a lexical dictionary into a non-negative matrix tri-factorisation model, using a simple rule-based polarity assignment. We refer the readers to Ding *et al.* [2008] and Taboada *et al.* [2011] for a detailed review on applying lexicon-based methods for sentiment analysis. Instead of using a lexicon, Jagarlamudi *et al.* [2012] used seeded words as lexical priors for semi-supervised topic modelling.

6.3 Opinion Mining Task

In this section, we describe the opinion mining task we are tackling. We then outline the major contributions of this work.

6.3.1 Problem Definition

Given a collection of documents (tweets), our first problem is to extract the *target–opinion* pairs from each document. A target–opinion pair $\langle w, o \rangle$ consists of two phrases: a *target* phrase w which is the object being described, and an *opinion* phrase o which is the description. The target phrases are usually nouns and the opinion phrases are usually adjectives. Examples include ⟨*picture quality, good*⟩, ⟨*iPhone app, expensive*⟩, and others. Note that a phrase can either be a *collocation* (multi-word phrase) or a single word. For simplicity, we will not distinguish between '*word*' and '*phrase*' in this dissertation, that is, a '*word*' can mean a *single-word* or a collocation.

Our next problem is to group the target–opinion pairs into clusters and identify the associated sentiments. The produced clusters should depend on the tweet corpus, as they should represent different aspects of the corpus. For example, given a tweet corpus which consists of various electronic products, we would like products that are different — such as mobile phones and computers — to be grouped into different clusters. Each target–opinion pair is assigned two latent labels, the first being *aspect*

[14] Each tweet is limited to at most 140 characters.

a indicating which cluster the pair belongs, the second label being *sentiment r*. The sentiment of a target–opinion pair refers to the polarity of the opinion phrase, which, in this chapter, can be *positive*, *neutral* or *negative*.

Finally, we would like to display a summary (high level view) of the obtained quadruples $\langle w, o, a, r \rangle$. There are many ways to do this, here we follow the standard topic modelling approach and display the top results. In brief, our task of opinion mining on tweets is to extract useful opinions and represent them in a format that is easy to digest. As an example, for a tweet corpus on electronic products, we would like to discover the users' opinions on certain products, such as the iPhones.

6.3.2 Major Contributions

We make two major contributions as follows: Firstly, *we design an LDA-based topic model* (TOTM) for performing aspect-based target–opinion analysis on product reviews from tweets. TOTM is novel in that it directly models the target–opinion interaction, giving significant improvement in opinion prediction. Existing aspect-based methods only model the interaction between aspects and sentiments, leaving the targets and opinions to be weakly associated through aspects and sentiments. Without this explicit modelling, the existing models failed to sensibly assign opinions to targets. To illustrate, from a restaurant review with *friendly staff* and *delicious cake*, existing LDA-based opinion models failed to recognise that the adjective '*friendly*' cannot be used to describe *cake*. Additionally, as mentioned in the introduction, TOTM makes use of available auxiliary variables in tweets (hashtags, mentions, emoticons and strong sentiment words) to improve aspect-based opinion analysis.

Secondly, *we propose a new formulation for incorporating a sentiment lexicon* into topic models. While existing methods adopt an *ad hoc* or ruled-based approach to incorporating sentiment prior, our formulation is novel in that it is learned automatically given the data. This is done robustly using a tuning hyperparameter that is optimised autonomously. The sentiment lexicon is used to adjust the opinion priors in order to improve sentiment analysis.

6.4 Interdependent LDA

The Interdependent LDA (ILDA) [Moghaddam and Ester, 2011], as illustrated in Figure 6.1, is an extension of LDA that performs aspect-based opinion analysis. It jointly models the aspect (a) and sentiment[15] (r) for each target–opinion pair $\langle w, o \rangle$ that are present in a document. We will assume that the sentiment r can only takes three labels, $\{-1, 0, 1\}$, which correspond to negative, neutral and positive sentiment respectively. However, we note that the sentiment variable r is an ordinal variable and is not restricted to just three values.

[15] Also known as *rating* in Moghaddam and Ester [2011].

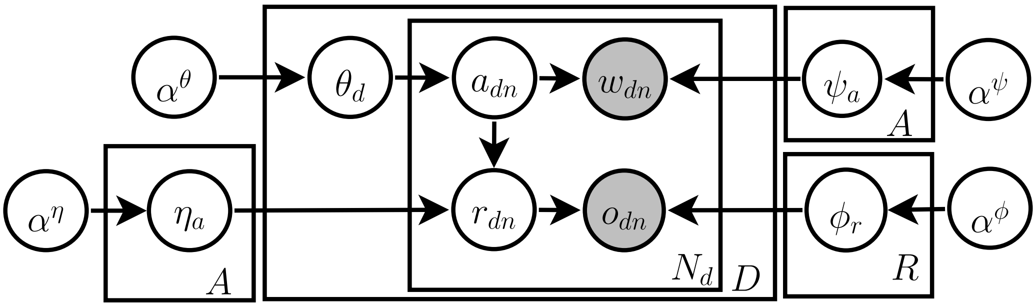


Figure 6.1: Graphical Model for the Interdependent LDA (ILDA). Given D document in a corpus and N_d target–opinion pairs $\langle w, o\rangle$ in document d, the *observed* variables (shaded) w and o are influenced by the latent labels a (aspect) and r (sentiment) respectively based on the aspect–target distributions ψ and the sentiment–opinion distributions ϕ. The interaction between aspect a and sentiment r is learned by the aspect-sentiment distribution η. The variable θ denotes the document–aspect distributions. All α are priors of the corresponding Dirichlet distributions.

The ILDA has the following generative process. For each document d in a corpus, we first sample a document–aspect distribution:

$$\theta_d \sim \text{Dirichlet}(\alpha^\theta)\,, \qquad \text{for } d = 1, \ldots, D\,. \tag{6.1}$$

Then, for each aspect a, we sample an aspect-sentiment distribution η_a and an aspect–target distribution ψ_a:

$$\eta_a \sim \text{Dirichlet}(\alpha^\eta)\,, \tag{6.2}$$

$$\psi_a \sim \text{Dirichlet}(\alpha^\psi)\,, \qquad \text{for } a = 1, \ldots, A\,. \tag{6.3}$$

Next, given each sentiment r, we sample a sentiment–opinion distribution:

$$\phi_r \sim \text{Dirichlet}(\alpha^\phi)\,, \qquad \text{for } r \in \{-1, 0, 1\}\,. \tag{6.4}$$

Finally, we model each target–opinion pair $\langle w_{dn}, o_{dn}\rangle$ and also the associated latent aspect a_{dn} and latent sentiment r_{dn}:

$$a_{dn} \,|\, \theta_d \sim \text{Discrete}(\theta_d)\,, \tag{6.5}$$

$$r_{dn} \,|\, a_{dn}, \eta \sim \text{Discrete}(\eta_{a_{dn}})\,, \tag{6.6}$$

$$w_{dn} \,|\, a_{dn}, \psi \sim \text{Discrete}(\psi_{a_{dn}})\,, \tag{6.7}$$

$$o_{dn} \,|\, r_{dn}, \phi \sim \text{Discrete}(\phi_{r_{dn}})\,, \qquad \text{for } n = 1, \ldots, N_d\,. \tag{6.8}$$

In the above description of the generative process, the variables α are the hyperparameters corresponding to symmetric Dirichlet distributions.

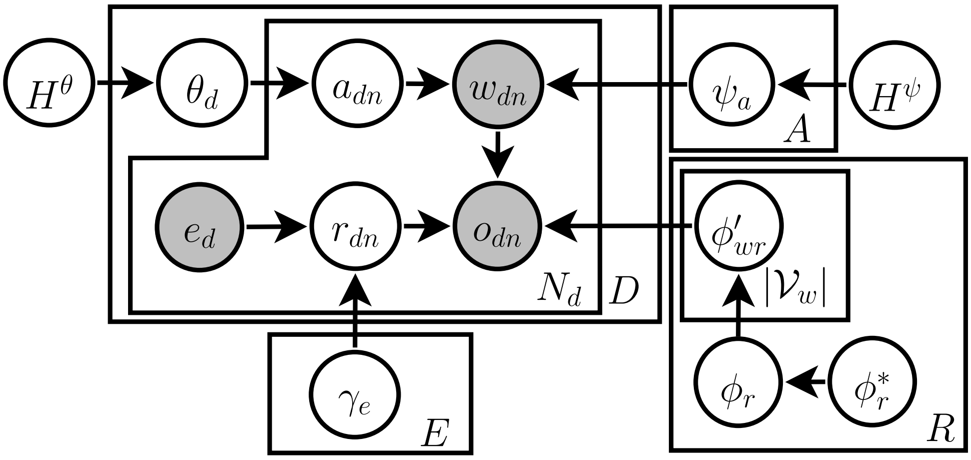


Figure 6.2: Graphical model for the Twitter Opinion Topic Model (TOTM). One aspect the TOTM is different to the ILDA is that it models the target–opinion interaction directly, as shown by the connection between w_{dn} and o_{dn}. Additionally, the TOTM utilises seen emoticons (e_d) in the document to enhance sentiment modelling. A hierarchical structure of the priors in the bottom right enables the incorporation of external sentiment lexicon.

ILDA models the sentiment conditionally on the aspect; and given the aspect and sentiment, the target word and opinion word are generated independently. Although such modelling is often adequate (since many of the opinion words can be applied generally to most target words), it fails to take into account that some opinion words are restricted to certain target words. For example, we can say that a phone has a *short battery life*, but not *short camera quality*. This shortcoming arises from the problem that opinion words are not tied with aspects (and hence target words).

6.5 Twitter Opinion Topic Model

Here we present TOTM for aspect-based opinion analysis on tweets. The graphical model is given in Figure 6.2. Contrary to ILDA, we do not model the aspect-sentiment distribution η. Instead, we model the target–opinion pairs directly. This allows us to better model the opinion words, and also provides us with a finer level of opinion analysis. For example, TOTM will be able to model that the word *'limited'* can describe *battery life* but is unlikely to be used to describe *charger*.

We introduce a variable e named *emotion indicator*, which detects the existence of emoticons and/or strong sentiment words in the documents. The strong sentiment words are hand-selected and represent words that are associated with a person's positive or negative feeling. The list of emoticons and strong sentiment words is presented in Table 6.1. We define e to be -1 when only negative emotion is observed and e to be 1 when only positive emotion is observed, otherwise we treat e as unob-

Table 6.1: List of Emoticons and Strong Sentiment Words. The positive tokens are associated with $e = 1$ while the negative tokens are linked to $e = -1$. These tokens are hand-picked from Wikipedia and online dictionaries. Note that the sentiment words include spelling variants of other regions (*e.g.*, American English) even though they are not explicitly listed here.

<table>
<tr><th>Type</th><th>Tokens</th></tr>
<tr><td>Positive Emoticons</td><td>:-) :o) :] :3 :c) :> =] 8) =) :} :^) ;) ;-) :-D ;-D :D 8-D 8D x-D xD X-D XD =-D =D =-3 =3 x3 B^D :)) :-)) \o/ *\0/* ^^ ^_^ (^_^)/ (^O^)/ (^o^)/ (^^)/ (^v^) (^u^) (^O^) (^o^))^o^(:} :-} =} C: (: (-:</td></tr>
<tr><td>Negative Emoticons</td><td>>:-(>:[:-(:-c :c :-< :< :-[:[:{ :-|| :@ >:(;(;-(:’-(:’(D8 D:< D: D; D= DX v.v D-’: (’_’) (/_;) (T_T) (;_;) (;_; (;_:) (;O;) (:_;) (ToT) T.T T_T t.t t_t u_u !_!):)-:)’:)-’:</td></tr>
<tr><td>Strong Positive Sentiment Words</td><td>love, like, happy, glad, delighted, content, cheerful, cheery, merry, joyful, jovial, jolly, gleeful, gratified, joyous, blessed, thrilled, elated, exhilarated, ecstatic, blissful, overjoyed, pleased, fortunate</td></tr>
<tr><td>Strong Negative Sentiment Words</td><td>hate, dislike, angry, sad, upset, unhappy, sorrowful, dismal, woeful, depressed, miserable, despairing, gloomy, broken-hearted, heartbroken, tragic, unfortunate, awful, sorrowful, grievous, traumatic, depressing, heartbreaking, agonised</td></tr>
</table>

served. Note that $e = 0$ would correspond to neutral emotion, but we have no such observation so this is not considered.

TOTM uses the *Pitman-Yor process* (PYP) [Teh, 2006b] to generate probability vector given another mean probability vector. The PYP is parameterised by a discount parameter α, a concentration parameter β, and a mean or base distribution H. Details on the PYP can be found in Section 3.3.2. The generative process of TOTM is as follows. First, we sample the document–aspect distribution θ_d for each document d,

$$\theta_d \sim \mathrm{PYP}(\alpha^{\theta_d}, \beta^{\theta_d}, H^{\theta}) \,, \qquad \text{for } \; d = 1, \ldots, D \,. \tag{6.9}$$

Here, H^{θ} is an arbitrary continuous distribution.

Second, we model the emotion-sentiment distribution γ_e by a Dirichlet distribution with asymmetric prior:

$$\gamma_e \,|\, e \sim \mathrm{Dirichlet}(q_e) \,, \qquad \text{for } \; e \in \{-1, 1\} \,. \tag{6.10}$$

The prior q_e is chosen such that $q_{-1} = (0.9, 0.05, 0.05)$ and $q_1 = (0.05, 0.05, 0.9)$.

Next, for the target words, we generate the aspect–target distribution ψ_a for each aspect a as follows:

$$\psi_a \sim \mathrm{PYP}(\alpha^{\psi_a}, \beta^{\psi_a}, H^{\psi}) , \qquad \text{for } a = 1, \ldots, A . \tag{6.11}$$

Here, H^{ψ} is a discrete uniform vector over the vocabulary of the target words ($\mathcal{V}_w$), that is, $H^{\psi} = (\ldots, \frac{1}{|\mathcal{V}_w|}, \ldots)$.

For the opinion words, we propose a novel hierarchical modelling that allows an opinion word to describe two different targets differently (*e.g.*, *short* for *processing time* is good but *short* for *battery life* is bad), while at the same time allows for sharing of the polarity of opinion words between targets. This is achieved by assigning common base distributions to the target–opinion distributions. So target–opinion distributions ϕ'_{wr} for different targets w share a common mean ϕ_r which itself is unknown so we sample it from a uniform base ϕ^*_r . More specifically, for each $r \in \{-1, 0, 1\}$ and $w \in \{1, \ldots, |\mathcal{V}_w|\}$, we generate ϕ'_{wr} as follows:

$$\phi^*_r = \left(\ldots, \frac{1}{|\mathcal{V}_o|}, \ldots\right) , \tag{6.12}$$

$$\phi_r \,|\, \phi^*_r \sim \mathrm{PYP}(\alpha^{\phi_r}, \beta^{\phi_r}, \phi^*_r) , \qquad \text{for } r \in \{-1, 0, 1\} , \tag{6.13}$$

$$\phi'_{wr} \,|\, \phi_r \sim \mathrm{PYP}(\alpha^{\phi'_{wr}}, \beta^{\phi'_{wr}}, \phi_r) , \qquad \text{for } w \in \{1, \ldots, |\mathcal{V}_w|\} , \tag{6.14}$$

where $\mathcal{V}_o$ is the vocabulary of the opinion words.

Finally, for each target–opinion pair $\langle w_{dn}, o_{dn} \rangle$ (indexed by n) in document d, we sample the respective aspect a_{dn} , sentiment r_{dn} and the target–opinion pair:

$$a_{dn} \,|\, \theta_d \sim \mathrm{Discrete}(\theta_d) , \tag{6.15}$$

$$r_{dn} \,|\, e_d, \gamma \sim \mathrm{Discrete}(\gamma_{e_d}) , \tag{6.16}$$

$$w_{dn} \,|\, a_{dn}, \psi \sim \mathrm{Discrete}(\psi_{a_{dn}}) , \tag{6.17}$$

$$o_{dn} \,|\, w_{dn}, r_{dn}, \phi' \sim \mathrm{Discrete}\big(\phi'_{w_{dn}, r_{dn}}\big) , \qquad \text{for } n = 1, \ldots, N_d . \tag{6.18}$$

We note that each PYP distribution is parameterised by its own set of hyperparameters, that is, β^{θ_d} differs for different document d. We present a list of variables associated with TOTM in Table 6.2. Also note that by modelling the target–opinion distribution explicitly, we have to store the information of the distribution for each target in the data, which is very large. In our implementation, we adopt a sparse representation for storing the counts associated with the target–opinion distributions. We find that each target word is only described by a limited number of opinion words in the data, which is less than 1 % of the words from the opinion word vocabulary.

In the next section, we propose a novel method to incorporate sentiment prior information for opinion analysis. It makes use of external sentiment lexicon that is publicly available.

Table 6.2: List of Variables for the Twitter Opinion Topic Model (TOTM).

Variable	Name	Description
a_{dn}	Aspect	Category label for target–opinion pair $\langle w_{dn}, o_{dn} \rangle$; also known as topic.
r_{dn}	Sentiment	Polarity of opinion phrase o_{dn} .
w_{dn}	Target	Observed target word or phrase that is being described at position n in document d.
o_{dn}	Opinion	Description of target w_{dn} .
e_d	Emotion indicator	Binary variable indicating positive or negative emotion in document d; can be unobserved.
ψ_a	Aspect–target distribution	Probability distribution in generating target words for aspect a.
ϕ'_{tr}	Opinion word distribution	Probability distribution in generating opinion words given target t and sentiment r.
ϕ_r	Opinion word distribution	Opinion prior for ϕ'_{tr} .
ϕ^*_r	Opinion word distributions	Opinion prior for ϕ_r .
θ_d	Document–aspect distribution	Probability distribution in generating aspects for document d.
γ_e	Sentiment distribution	Probability distribution in generating sentiments for emotion indicator e.
$\alpha^{\mathcal{N}}$	Discount	Discount parameter for PYP $\mathcal{N}$.
$\beta^{\mathcal{N}}$	Concentration	Concentration parameter for PYP $\mathcal{N}$.
$H^{\mathcal{N}}$	Base distribution	Base distribution for PYP $\mathcal{N}$.

6.6 Incorporating Sentiment Prior

He [2012] proposed a simple yet effective way to incorporate sentiment prior information into LDA by directly modifying the Dirichlet prior based on available sentiment lexicons. Naming her model LDA-DP (LDA with Dirichlet Prior modified), she replaces the topics in LDA by latent sentiment labels and allows the word priors to be custom probability distributions. The generative process of LDA-DP is identical to LDA and hence omitted.

In LDA-DP, the word distribution ϕ_r is Dirichlet distributed with the parameter $\alpha_r \lambda_r$, where λ_r is a vector of length $|\mathcal{V}_o|$, and $r \in \{-1, 0, 1\}$ is the sentiment label corresponding to negative, neutral and positive sentiment, respectively.[16] The λ_{rv} is initialised to be $1/3$, and subsequently updated if the sentiment lexicon contains word v. In this case, λ_{rv} takes the value of 0.9 if the sentiment of word v matches r, and takes the value of 0.05 otherwise:

$$\lambda_{rv} = \begin{cases} 0.9 & \text{if Sentiment}(v) = r \\ 0.05 & \text{otherwise} \end{cases}, \qquad \text{for } v = 1, \ldots, |\mathcal{V}_o| \, . \tag{6.19}$$

Motivated by this, but not wishing to be required to give the exact strength by which the dictionary affects probabilities, we instead propose a novel formulation that automatically learns and updates itself. We assume that a sentiment lexicon is available and it provides sentiment scores for opinion words. Additionally, we assume that the sentiment score S_v returned from the sentiment lexicon takes a negative value when v has a negative sentiment, a positive value when v has a positive sentiment, and 0 when v is neutral.[17]

Sentiment lexicons that are freely available online include the SentiWordNet [Baccianella *et al.*, 2010], SentiStrength [Thelwall *et al.*, 2010], MPQA Subjectivity lexicon [Wilson *et al.*, 2005] and others. SentiStrength is developed from MySpace[18] text data by the Statistical Cybermetrics Research Group from the University of Wolverhampton, UK. Since the SentiStrength lexicon is constructed for informal text, we use it to extract sentiment information for TOTM. The sentiment score S_v from SentiStrenth ranges from -5 to $+5$, which conforms to our assumption. We assume that $S_v = 0$ for unlisted words.

Additionally, we make use of the SentiWordNet 3.0 lexicon to evaluate TOTM. SentiWordNet is built on WordNet [Fellbaum, 1998] by researchers from Italy. We note that SentiStrength and SentiWordNet are developed independently by different teams using different methods. Thus we claim it is fair and unbiased to use one lexicon for training and the other one for evaluation.

Our formulation is as follows, we introduce a tunable parameter b that controls the strength of the prior, and replace the prior ϕ_r^* (in the context of TOTM) by

$$\phi_{rv}^* \propto (1 + b)^{X_{rv}} \, , \tag{6.20}$$

where $b > 0$ and hence $\phi_{rv}^* > 0$. Here, X_{rv} is the score of word v for sentiment r,

[16] We redefined the original sentiment labels in He [2012] for consistency.

[17] We can simply normalise the score to conform to this assumption, when the assumption is not met.

[18] MySpace is a social networking website similar to Facebook.

which is defined as

$$X_{rv} = \begin{cases} S_v & \text{if } r = 1 \ \ \text{(positive)} \\ -|S_v| & \text{if } r = 0 \ \ \text{(neutral)} \\ -S_v & \text{if } r = -1 \ \ \text{(negative)} \ . \end{cases} \tag{6.21}$$

Note that although there are multiple ways to formulate the prior, we choose the above formulation due to its simplicity and intuitiveness. We can see that a positive X_{rv} boosts the probability of word v while a negative X_{rv} diminishes it. Also, this formulation ensures the positivity of the prior, which can be difficult to achieve if we were to use other formulations such as a polynomial function.

Even though b is a tunable parameter, we do not need to manually tune it. We propose a flexible way to learn the hyperparameter b from its posterior distribution, thus relieving us from choosing the value for b, which can be difficult (the value of b should depend on the sentiment scores of the lexicon). The learning of the hyperparameter b is detailed in Section 6.7.2.

6.7 Inference Techniques

In this section, we first discuss the collapsed Gibbs sampler for TOTM, and continue with the sampling procedure of the hyperparameters. We note that this inference procedure is developed upon the learning method described in Section 5.4.

6.7.1 Collapsed Gibbs Sampling for TOTM

As discussed in Section 5.3, the key to Gibbs sampling with PYPs is to marginalise out the probability vectors in the model and record the associated customer counts and table counts. Here, we adopt the model representation in Section 5.3 and marginalise out the variables θ, γ, ψ, ϕ', ϕ, and ϕ^*. As previously defined, c denotes the customer counts and t denotes the table counts.

The variables **A**, **R**, **W**, and **O** represent a collection of relevant variables, as mentioned in Section 1.4. We also denote Ξ as the set of all hyperparameters (including b). The posterior likelihood of the model can be written — in terms of $f(\cdot)$ from Equation (5.9) — as

$$\begin{aligned} p(\mathbf{A}, \mathbf{R}, \mathbf{T}, \mathbf{C} \,|\, \mathbf{W}, \mathbf{O}, \Xi) \propto & \left(\prod_{d=1}^{D} f(\theta_d) \right) \left(\prod_{r=-1}^{1} f(\phi_r) \left[\prod_{w=1}^{|\mathcal{V}_w|} f(\phi'_{wr}) \right] \left[\prod_{v=1}^{|\mathcal{V}_o|} (\phi^*_{rv})^{t_v^{\phi_r}} \right] \right) \\ & \left(\prod_{e \in \{-1,1\}} f(\gamma_e) \prod_{u=-1}^{1} (q_{eu})^{t_u^{\gamma_e}} \right) \left(\prod_{a=1}^{A} f(\psi_a) \prod_{w=1}^{|\mathcal{V}_w|} \left(\frac{1}{|\mathcal{V}_w|} \right)^{t_w^{\psi_a}} \right) . \end{aligned} \tag{6.22}$$

As detailed in Section 5.4, the collapsed Gibbs sampler consists of decrementing counts associated with a word, sampling the respective new latent values for the word, and incrementing the respective counts. For TOTM, we alternatingly sample a new aspect a and sentiment r. As before, the conditional posteriors are ratio of the posterior likelihoods, which can further be simplified to a ratio of modularised likelihoods like those in Equation (5.17).

The conditional posterior for aspect a_{dn} can be derived as

$$p(a_{dn}, \mathbf{T}, \mathbf{C} \,|\, \mathbf{A}^{-dn}, \mathbf{R}, \mathbf{W}, \mathbf{O}, \mathbf{C}^{-dn}, \mathbf{T}^{-dn}, \Xi) = \frac{p(\mathbf{A}, \mathbf{R}, \mathbf{T}, \mathbf{C} \,|\, \mathbf{W}, \mathbf{O}, \Xi)}{p(\mathbf{A}^{-dn}, \mathbf{R}, \mathbf{T}^{-dn}, \mathbf{C}^{-dn} \,|\, \mathbf{W}, \mathbf{O}, \Xi)}\,, \tag{6.23}$$

while the conditional posterior for sentiment r_{dn} is

$$p(r_{dn}, \mathbf{T}, \mathbf{C} \,|\, \mathbf{A}, \mathbf{R}^{-dn}, \mathbf{W}, \mathbf{O}, \mathbf{C}^{-dn}, \mathbf{T}^{-dn}, \Xi) = \frac{p(\mathbf{A}, \mathbf{R}, \mathbf{T}, \mathbf{C} \,|\, \mathbf{W}, \mathbf{O}, \Xi)}{p(\mathbf{A}, \mathbf{R}^{-dn}, \mathbf{T}^{-dn}, \mathbf{C}^{-dn} \,|\, \mathbf{W}, \mathbf{O}, \Xi)}\,. \tag{6.24}$$

Here, the superscript $\square^{-dn}$ indicates that the target–opinion pair $\langle w_{dn}, o_{dn} \rangle$ and their associated variables are removed from the respective sets.

6.7.2 Hyperparameter Sampling

During inference, we sample the hyperparameters of the PYP using an auxiliary variable sampler [Teh, 2006a], for details, see Section 5.4.3. Moreover, we propose a novel method to update the hyperparameter b, which controls the strength of the sentiment prior. Instead of sampling the hyperparameter b (*e.g.*, using the slice sampler [Neal, 2003]), we adopt an optimisation approach since the posterior of b is highly concentrated in a small region (thin-tailed). The posterior density is given by the following equation, subject to a normalisation constant.

$$p(b \,|\, c) \propto p(b) \prod_{r=-1}^{1} \prod_{v=1}^{|\mathcal{V}_o|} \left(\frac{(1+b)^{X_{rv}}}{\sum_i (1+b)^{X_{ri}}} \right)^{c_{rv}}\,, \tag{6.25}$$

where c_{rv} is the number of times an opinion word v is assigned a sentiment r,[19] and $p(b)$ is the hyperprior of b. We assume a weak hyperprior for b, so

$$b \sim \text{Gamma}(1, 1)\,, \tag{6.26}$$

$$p(b) \propto e^{-b}\,. \tag{6.27}$$

During inference, we update b to its *maximum a posteriori probability* estimate using a gradient ascent algorithm. We optimise for its log posterior, $l(b) := \log p(b \,|\, c)$,

[19] In fact, c_{rv} is equal to $t_v^{\phi_r}$.

Algorithm 6.1 Gradient Ascent Optimisation for Hyperparameter b

```
1. Given an initial value for $b = b_0$, evaluate the gradient $l'(b_0)$.
2. Given a learning rate $\tau$, update $b$ to $b_i = b_{i-1} + \tau \times l'(b_{i-1})$, if the new log
   posterior $l(b_i)$ is lower than $l(b_{i-1})$, we halve the learning rate: $\tau := \tau/2$ .
3. Repeat Step 2 until $b$ converges.
```

Algorithm 6.2 Collapsed Gibbs Sampling for TOTM

```
1. Initialise the model by assigning a random aspect to each target–opinion pair,
   sampling the sentiment label, and building the relevant customer counts $c_k^{\mathcal{N}}$
   and table counts $t_k^{\mathcal{N}}$ for all PYP $\mathcal{N}$.
2. For each document $d$, and
   (a) For each target phrase $w_{dn}$, perform the following:
       i. Decrement counts associated with $w_{dn}$.
       ii. Sample a new aspect $a_{dn}$ and corresponding customer counts and table
           counts from Equation (6.23).
       iii. Increment the associated counts for the new $a_{dn}$.
   (b) For each opinion phrase $o_{dn}$, perform the following:
       i. Decrement counts associated with $o_{dn}$.
       ii. Sample a new sentiment $r_{dn}$ and corresponding customer counts and
           table counts from Equation (6.24).
       iii. Increment the associated counts for the new $r_{dn}$.
3. Update the hyperparameters $\beta$ and $b$.
4. Repeat Steps 2–3 until the model converges or when a fixed number of itera-
   tions is reached.
```

since log is an increasing function. The gradient of the log posterior is derived as

$$l'(b) = \frac{1}{1+b} \sum_{r=-1}^{1} \sum_{v=1}^{|\mathcal{V}_o|} c_{rv} \big(X_{rv} - \mathbb{E}_{\phi_r^*}[X_r] \big) + \rho'(b)\,, \tag{6.28}$$

where $\mathbb{E}_{\phi_r^*}[X_r]$ is the expected value of X_r under the probability distribution ϕ_r^*, and $\rho'(b)$ is the derivative of $\log p(b)$. We summarise the gradient ascent algorithm in Algorithm 6.1. We refer the reader to Appendix A.1 for the gradient derivation.

We summarise the collapsed Gibbs sampler in Algorithm 6.2. In the next section, we describe the data used for evaluating the TOTM.

Table 6.3: Keywords for querying the electronic product dataset. Note that some keywords are listed in multiple categories as they are not mutually exclusive.

Categories	Query Words
Mobile phones	iphone, blackberry, nokia, palmpre, sony, motorola, phone, samsung, lg, scanner, android, ios, apple
Computers	sony, dell, lenovo, toshiba, acer, asus, macbook, hp, alienware, laptop, tablet, netbook, ipad, ipod, printer, panasonic, epson, samsung, ibm, sony, microsoft, computer, windows, operatingsystem, apple
Cameras	sony, canon, nikon, camera, panasonic, epson, samsung, lg, fujitsu, kodak
Printers/ Scanners	sony, canon, nikon, dell, lenovo, toshiba, hp, printer, panasonic, epson, samsung, kyocera, lg, scanner, kodak
Gaming	xbox, playstation, wii, nintendo, gameboy, sega, squareenix

6.8 Data

For experiments, we perform aspect-based opinion analysis on tweets, which are characterised by their limited 140 characters text. From the *Twitter 7* dataset[20] [Yang and Leskovec, 2011], we queried for tweets that are related to electronic products such as *camera* and *mobile phones* (see the list of the query words in Table 6.3). We then remove non-English tweets with *langid.py* [Lui and Baldwin, 2012]. Moreover, since most spam tweets contain a URL, we adopt a conservative approach to remove spam by discarding tweets containing URLs. This results in a dataset of about 9 million tweets, which we name as the electronic product dataset.

Due to the lack of sentiment labels on the electronic product dataset, we make use of the Sentiment140 (Sent140) tweets[21] [Go *et al.*, 2009] for sentiment classification. Each Sent140 tweet contains a sentiment label (positive or negative) that are determined by emoticons. The whole corpus contains 1.6 million tweets, with half of them labelled as positive and the other half as negative.

In addition, we also use the SemEval 2013 dataset[22] [Nakov *et al.*, 2013] for evaluation. SemEval tweets are annotated on Mechanical Turk, which arguably provides better sentiment labels compared to Sent140. Since annotation is expensive, SemEval has only 6,322 tweets.

[20] http://snap.stanford.edu/data/twitter7.html (last accessed 11 June 2014)

[21] http://help.sentiment140.com/home (last accessed 11 June 2014)

[22] http://www.cs.york.ac.uk/semeval-2013/task2/ (last accessed 11 June 2014)

6.8.1 Data Preprocessing

Here, we describe the preprocessing steps that we apply to tweets. Firstly, we apply Twitter NLP [Owoputi *et al.*, 2013], a state-of-the-art tool for part-of-speech (POS) tagging on tweets. We then apply word normalisation to clean up the tweets. We make use of the lexical normalisation dictionary[23] from Han *et al.* [2012], but modify it such that proper nouns are not normalised. For instance, words like 'iphone' and 'xbox' are not normalised, since they are the targets we are interested in. We perform normalisation after POS tagging since tweets normalisation degrades the performance of Twitter NLP [Han *et al.*, 2013].

Next, we proceed to extract target–opinion pairs from the datasets. Following Moghaddam and Ester [2012], we apply the Stanford Dependency Parser [De Marneffe *et al.*, 2006] to extract the dependency relations that will be used to form the target–opinion pairs. However, our approach is slightly different: we do not use the *Direct Object* (*dobj*) relation to obtain a target–opinion pair. For example, the sentence "*I like the perfect picture quality*" gives '*dobj*(like, picture quality)' and '*amod*(picture quality, perfect)', resulting in two target–opinion pairs, ⟨*picture quality, like*⟩ and ⟨*picture quality, perfect*⟩. We drop the target–opinion pair associated with *dobj* as it is not suitable for target–opinion pair, instead, we use the *dobj* relation for the emotion indicator variable. Note that we use the *caseless English model* in the Stanford Dependency Parser, which works better for tweets. Additionally, since standard NLP tools perform less optimally on tweets [Ritter *et al.*, 2011], we use the POS tagging from Twitter NLP to clean up the target–opinion pairs. We note that negations like '*not*' are captured as dependency relations, the negated words are then treated as new words with the prefix '*not_*'.

We determine the emotion indicator variable *via* the existence of emoticons, strong sentiment words and/or the *dobj* relations in each tweet. We simply set the emotion indicator to -1 (negative) or 1 (positive) as long as the indicators agree with one another, and unobserved otherwise. The list of emoticons is compiled from Wikipedia.[24] The emoticons and strong sentiment words are presented in Table 6.1. For Sent140 and SemEval tweets, we replace the unobserved emotion indicator by their sentiment label.

We then perform tweet aggregation, which is found to provide significant improvement for LDA [Mehrotra *et al.*, 2013]. We group tweets that contain the same hashtag (word prefixed with # symbol) or same mention (word prefixed with @ symbol) into a single document. This allows co-occurrence within the same *tags* (our abbreviation for hashtags and mention) to be used by topic models.[25] Grouping tweets also allows us to summarise the results for each tag, giving us a better opin-

[23] http://people.eng.unimelb.edu.au/tbaldwin/#resources (last accessed 11 June 2014)

[24] http://en.wikipedia.org/wiki/Kaomoji and http://en.wikipedia.org/wiki/List_of_emoticons (last accessed 11 June 2014)

[25] Tweets with multiple tags are replicated into multiple pseudo documents.

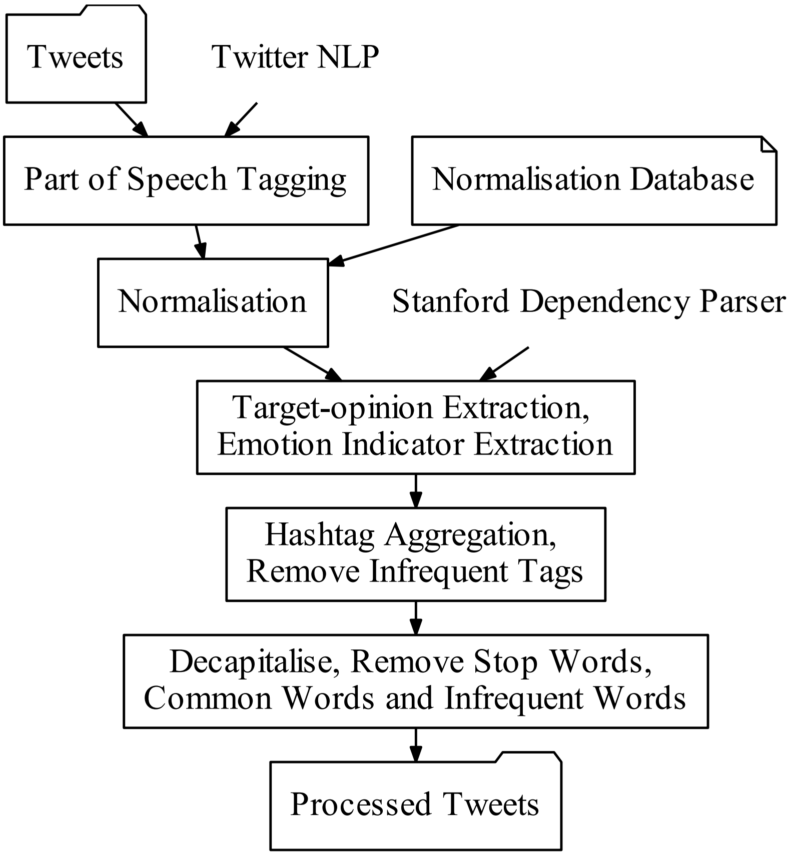

Tweets
Twitter NLP
Part of Speech Tagging
Normalisation Database
Normalisation
Stanford Dependency Parser
Target-opinion Extraction,
Emotion Indicator Extraction
Hashtag Aggregation,
Remove Infrequent Tags
Decapitalise, Remove Stop Words,
Common Words and Infrequent Words
Processed Tweets

Figure 6.3: Preprocessing pipeline for tweets.

ion overview (see Section 6.9.3 for examples). Additionally, we discard tags that occur infrequently. We note that although tweets are merged to form a larger document, the emotion indicator (variable e) is observed and stored for each individual tweet (rather than the merged document). This prevents the emotion indicator from being lost through merging.

Finally, we perform other standard preprocessing techniques to topic modelling, this consists of decapitalising the words, removing stop words[26] and discarding commonly occurred words and infrequent words. For example, we define the common words as words that appear in at least 90 % of the documents, and infrequent words as words that appear less than 50 times in the corpus. We randomly split the data into 90 % training set and 10 % test set for evaluation. A summary of the preprocessing pipeline is displayed in Figure 6.3.

[26] The stop words list is obtained from MALLET [McCallum, 2002].

Table 6.4: Corpus Statistics for the Electronic Product, Sent140 and SemEval tweets, showing the number of tweets, the number of target–opinion pairs extracted per tweet, proportion of tweets with observed emotion indicator, the size of target word vocabulary, and the size of opinion word vocabulary.

	Electronic Product	Sent140	SemEval
Number of tweets	∼9M	1.6M	6322
Target–opinion pairs per tweet	0.69	0.41	0.47
Percentage of tweets with observed e	17.9	100	57.5
Target vocabulary size ($\|\mathcal{V}_w\|$)	4402	1050	1875
Opinion vocabulary size ($\|\mathcal{V}_o\|$)	25188	8599	813

6.8.2 Corpus Statistics

On average, we found that there are 0.69 target–opinion pair extracted per electronic product tweet. Out of the electronic product tweets that contain at least one target–opinion pair, 17.9 % of them contain an emotion indicator. After preprocessing, the number of unique target word tokens in the electronic product tweets is 4,402, while the number of unique opinion word tokens is 25,188. We present a summary of the corpus statistics for all datasets in Table 6.4.

For the electronic product tweets, the top tags are #apple, #phone, #iphone, #computer, and #laptop. We note that tags are associated with products, brands or companies. For example, #playstation and #xbox are associated with gaming products, while #sony and #canon are associated with companies. In Section 6.9.3, we show that aggregating hashtags allows us to have a more focussed view on certain products or companies, as well as facilitating comparison between these products or companies side-by-side.

6.9 Experiments and Results

In this section, we demonstrate the usefulness of TOTM for opinion mining. We evaluate TOTM quantitatively against ILDA and LDA-DP in terms of goodness-of-fit and sentiment classification. To compare the effectiveness of various sentiment lexicons, we propose a novel sentiment metric to evaluate the sentiment–opinion word distributions ϕ. Qualitatively, we employ TOTM for the task of opinion mining from the electronic product tweets, and demonstrate that we are able to extract various useful opinions on technological products such as the iPhone. Note that we do not compare against other models such as MG-LDA and ASUM, since these models do not perform target-based opinion analysis, and thus not directly comparable.

6.9.1 Experiment Settings

For all the experiments, we initialise the hyperparameters of PYP to $\alpha = 0.7$ and $\beta = 0.1$. The discount of 0.7 is chosen to induce power-law behaviour on the word distributions. For the sentiment hyperparameter b, we initialise it to $b = 10$ as we find that this starting value works well. Note that these hyperparameters are optimised automatically as discussed in Section 6.7.2.

To determine the optimal number of latent aspects (A) for ILDA, we set aside 5 % of the training data as development set, and select A (tested in increment of 10) such that perplexity of the development set is minimised. For a fair comparison between TOTM and ILDA, we cap the maximum number of aspects of TOTM to be that of ILDA. Our experiment finds that the number of aspects in TOTM always converges to the cap. We note that LDA-DP has only three fixed '*topics*', which is the number of sentiments.

During inference, we run the collapsed Gibbs sampling algorithm until the convergence criteria is satisfied, defined by which the training log likelihood not differing by more than 0.1 % in ten consecutive iterations. Empirically, we find that all experiments converge within 500 iterations, indicating a good Gibbs sampling algorithm. We refer the readers to Section 6.10.1 for a more detailed discussion on analysing convergence.

6.9.2 Quantitative Evaluations

Here we present the quantitative results of TOTM on model fitting and sentiment classification, comparing against ILDA and LDA-DP. Additionally, we also test the effectiveness of the sentiment lexicons through the sentiment prior evaluation.

6.9.2.1 Goodness-of-fit Test

As discussed in Section 5.5.2, we compute the perplexity of the test set to measure how well the model fits the data. Since aspect-based opinion analysis deals with two types of vocabulary, we compute the perplexity for both target words **W** and opinion words **O**, in this case:

$$\text{perplexity}(\mathbf{W} \mid \theta, \psi) = \exp\left(- \frac{\sum_{d=1}^{D} \sum_{n=1}^{N_d} \log p(w_{dn} \mid \theta_d, \psi)}{\sum_{d=1}^{D} N_d} \right), \qquad (6.29)$$

$$\text{perplexity}(\mathbf{O} \mid \mathbf{W}, \mathbf{E}, \gamma, \phi') = \exp\left(- \frac{\sum_{d=1}^{D} \sum_{n=1}^{N_d} \log p(o_{dn} \mid w_{dn}, e_d, \gamma, \phi')}{\sum_{d=1}^{D} N_d} \right), \qquad (6.30)$$

noting that N_d is the number of the target–opinion pairs in document d. We also

Table 6.5: Perplexity results on all datasets. The TOTM achieves significant (at 5 % significance level) perplexity reduction on the opinion words, leading to overall perplexity reduction. We note that the lower the perplexity, the better the model fitting, as discussed in Section 5.5.2.

Dataset	Models	Target Perplexity	Opinion Perplexity	Overall Perplexity
Electronic Product	LDA-DP	N/A	510.15 ± 0.08	N/A
	ILDA	594.81 ± 13.61	519.84 ± 0.43	556.03 ± 6.22
	TOTM	592.91 ± 13.86	**137.42** ± 0.28	**285.42** ± 3.23
Sent140	LDA-DP	N/A	329.92 ± 16.58	N/A
	ILDA	567.22 ± 16.31	306.79 ± 0.15	417.12 ± 6.12
	TOTM	530.08 ± 5.23	**93.89** ± 0.41	**223.09** ± 0.63
SemEval	LDA-DP	N/A	688.54 ± 62.17	N/A
	ILDA	2695.39 ± 65.33	433.20 ± 1.50	1080.51 ± 13.75
	TOTM	2725.51 ± 71.88	**249.04** ± 4.09	**823.74** ± 7.68

compute the overall perplexity, which is given by

$$\text{perplexity}(\mathbf{W},\mathbf{O}\,|\,\theta,\mathbf{E},\gamma,\psi,\phi') = \exp\left(-\frac{\sum_{d=1}^{D}\sum_{n=1}^{N_d}\log p(w_{dn},o_{dn}\,|\,\theta,e_d,\gamma,\psi,\phi')}{2\sum_{d=1}^{D}N_d}\right). \tag{6.31}$$

We present the perplexity result (the lower the better) for the electronic product, Sent140, and SemEval tweets in Table 6.5. From the perplexity results, it is clear that modelling the target–opinion pairs directly leads to significant improvement on opinion words perplexity and hence the overall perplexity. Note that LDA-DP only models the opinion words, thus we can only compare the perplexity for opinion words, we can see that its result is comparable to that of ILDA.

6.9.2.2 Sentiment Classification

Here, we perform a classification task to predict the polarity of the test data for Sent140 and SemEval tweets. We determine the polarity of a test document d simply by selecting the polarity r that is more probable (higher likelihood):

$$\text{polarity}(d) = \arg\max_{r=\{-1,1\}} \prod_{n=1}^{N_d} \phi_{r,o_{dn}}\,. \tag{6.32}$$

For simplicity, the sentiment classification is a binary classification task, as such, we do not include neutral tweets from SemEval data during evaluation. Note that

Table 6.6: Sentiment classification results for Sent140 and SemEval tweets. We can see that TOTM outperforms ILDA and LDA-DP on all aspects. Here, the results are between 0 and 100, with larger values indicating better performance.

Dataset	Models	Accuracy [%]	Precision [%]	Recall [%]	F measure [%]
Sent140	LDA-DP	57.3	56.1	90.1	69.2
	ILDA	54.1	56.9	55.3	55.9
	TOTM	**65.0**	**61.7**	**90.2**	**73.3**
SemEval	LDA-DP	52.1	65.0	58.3	61.4
	ILDA	46.8	60.7	53.6	56.3
	TOTM	**73.3**	**84.0**	**74.9**	**79.0**

Sent140 data does not have neutral tweets.

We present the classification *accuracy*, *precision*, *recall* and the *F measure* [Suominen *et al.*, 2008] in Table 6.6. We can see that TOTM outperforms LDA-DP and ILDA on both datasets, suggesting that our prior formulation is more appropriate than that of LDA-DP. We can also see that LDA-DP gives a better sentiment classification compared to ILDA, which does not incorporate any prior information. Note that the classification result for SemEval data is better than that of Sent140. We conjecture that this is because the sentiment labels in Sent140 are obtained from the emoticons, which are noisy in nature; while the sentiment labels for SemEval data is annotated, thus can be predicted more accurately.

6.9.2.3 Evaluating the Sentiment Prior

We propose a novel method to evaluate the learned sentiment–opinion word distributions ϕ by using another sentiment lexicon. We use the SentiWordNet lexicon for evaluation, noting that the lexicon used during training is the SentiStrength lexicon.

Unlike SentiStrength, the SentiWordNet lexicon provides two values for each word. We name them the positive affinity Z_v^+ and negative affinity Z_v^- for a given word v, they ranged from 0 to 1. For example, the word '*active*' has a positive affinity of 0.5 and a negative affinity of 0.125; while '*supreme*' has a positive affinity of 0.75 and a negative affinity of 0.

Given the affinities, we propose the following sentiment score to evaluate the opinion word distributions:

$$Score(\phi_r, Z) = E_{\phi_r}[Z] = \sum_{v=1}^{|\mathcal{V}_o|} Z_v \, \phi_{rv} \,, \tag{6.33}$$

where Z can either be Z^+ or Z^-, the positive or negative affinity. Note the sentiment score is also the expected sentiment under the opinion word distribution.

Table 6.7: Sentiment Evaluations for the Sentiment Priors (in unit of **0.01**). The results, where the higher the better, suggest that incorporating sentiment prior into modelling is important.

Dataset	Lexicon	Negative Affinity	Positive Affinity
Electronic	No lexicon	17.82 ± 1.26	17.39 ± 0.45
	MPQA	**23.91** ± 0.49	31.96 ± 0.09
	SentiStrength	23.19 ± 0.08	**35.69** ± 0.33
Sent140	No lexicon	22.63 ± 0.96	32.31 ± 1.98
	MPQA	24.10 ± 0.49	**42.65** ± 1.02
	SentiStrength	**24.29** ± 1.07	41.26 ± 1.53
SemEval	No lexicon	15.24 ± 1.45	21.03 ± 3.85
	MPQA	16.88 ± 0.31	29.47 ± 0.99
	SentiStrength	**16.94** ± 0.78	**32.17** ± 2.07

Here, we evaluate ϕ_{-1} with negative affinity Z^{-} and ϕ_1 with positive affinity Z^{+}. We compare the sentiment scores between the cases when a sentiment lexicon is used and when it is not. Additionally, we also make use of the MPQA Subjectivity lexicon for sentiment prior (during training) and compare the sentiment evaluation against the SentiStrength lexicon. We present the result in Table 6.7. As we can see, it is clear that incorporating prior information results in huge improvement in the sentiment score. Also, the priors for SentiStrength are slightly better than MPQA, on average. We note that optimising the hyperparameter b is very important, as it relieves us from tuning the hyperparameter manually. To illustrate, the optimised b converges to 2.59 on the electronic product tweets, while on Sent140 and SemEval dataset, the b converges to 1.85 and 0.71 respectively. We also find that, in our tests, an incorrectly chosen b can lead to a bad result.

6.9.3 Qualitative Analysis and Applications

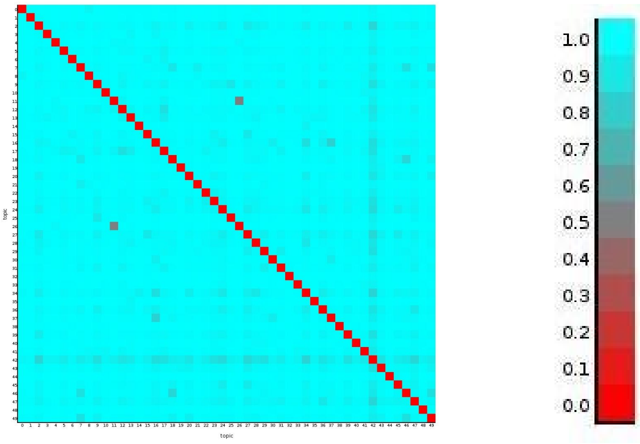
In addition to quantitative evaluations, qualitative analysis is important too in assessing topic models. Here, we examine the quality of the learned distributions and demonstrate the usefulness of TOTM on opinion mining.

6.9.3.1 Analysing Word Distributions

First, we inspect the clustering of target words by the TOTM and the ILDA, noting that the LDA-DP does not model the target words. As mentioned in Section 5.5.3, the Hellinger distance is commonly used to measure the dissimilarity between two probability distributions. We calculate the pairwise Hellinger distance between each aspect–target word distribution and found that the aspects are distinctive. The

Table 6.8: Top target words for the electronic product tweets learned by TOTM. The aspect labels are manually assigned based on the target words. We find that the top target words form coherent topics for each aspect.

Aspect (a)	Target Words (w)
Camera	camera, pictures, video camera, shots
Apple iPod	ipod, ipod touch, songs, song, music
Android phone	android, apps, app, phones, keyboard
Macbook	macbook, macbook pro, macbook air
Nintendo games	nintendo, games, game, gameboy



(a) Heat Map

(b) Legend

Figure 6.4: Pairwise Hellinger distances for the 50 aspect–target word distributions learned by TOTM on the electronic product tweets. The heat map shows that most of the pairwise Hellinger distances are high, indicating that the aspects are distinct.

Hellinger distances between all pairs of aspect–target word distributions from the TOTM are displayed as a heat map in Figure 6.4, we can see that the distances between the aspects are high, indicating that there is no duplicated aspect. We note that the heat map for the ILDA is similar and hence not presented here. In Table 6.8, we display an extract of the top target words which are learned by the TOTM from the electronic product tweets. Our empirical examination on the aspect–target word distributions suggest that both the TOTM and the ILDA perform well in clustering the target words.

We then look at the opinion word distributions. In ILDA and LDA-DP, the opinion words are generated conditioned on the latent sentiment labels, meaning that the opinion word is assumed to be independent to the target word given the sentiment;

Table 6.9: Opinion analysis of target words with TOTM on the electronic product tweets. Words in **bold** are more specific and can only describe certain targets. Here, we use red colour for positive sentiment and blue colour for negative sentiment.

Target (w)	Sentiment (r)	Opinions (o)
phone	$+1$	**mobile smart** good great f***ing
	-1	**dead** damn stupid bad crazy
battery life	$+1$	good **long** great **7hr ultralong**
	-1	terrible poor bad horrible **non-existence**
game	$+1$	great good awesome favorite **cat-and-mouse**
	-1	**addictive** stupid free **full addicting**
sausage	$+1$	**hot grilled** good **sweet** awesome
	-1	silly **argentinian cold** huge stupid

while in TOTM, the opinion word distributions are modelled given the sentiment and the observed target word. The advantage of TOTM over ILDA and LDA-DP in modelling the opinion words is that it allows us to analyse the opinions in a finer grained view. For instance, we can display a list of positive and negative opinions associated to a certain target word; an extract of this result is presented in Table 6.9, in which we pick a few distinctive target words to show their opinion word distributions. As we can see from Table 6.9, despite some opinion words can generally be applied to most target words (*e.g.*, good, bad), the highlighted words are more descriptive (*e.g.*, addictive, fried, grilled) and can only be applied to certain target words. Such result cannot be achieved with ILDA or LDA-DP.

6.9.3.2 Comparing Opinions on Brands with TOTM

We present an application of comparing opinions on entities or products using the TOTM. Since entities and products are frequently quoted with tags, we can compare them directly by looking at the opinions associated with each tag.

We present an extract of the opinion comparison between three brands (Canon, Sony and Samsung) in Table 6.10. It shows that we can have a high level comparison of the camera product between these three brands. For the phone product, there are only comparison between Sony and Samsung, since Canon does not manufacture phones (or no tweet on such topic is found).

6.9.3.3 Extracting Contrastive Opinions on Products

Although the above comparison is useful for providing a high level summary, it is also important to inspect the original tweets as they provide opinions in greater details. We use the TOTM to extract tweets containing people's opinions on iPhone.

Table 6.10: Aspect-based opinion comparison between Sony, Canon and Samsung. This extract shows two common aspects of the three brands, namely "camera" and "phone". The comparison allows us to quickly view the opinions of Twitter users on their products. Again, positive sentiment is shown in red while the negative is in blue colour.

Brands	Sentiment	Aspects / Targets' Opinions Camera	 Phone
Canon	+1	*camera* → great compact amazing *pictures* → great nice creative	
	−1	*camera* → expensive small bad *lens* → prime cheap broken	
Sony	+1	*photos* → great lovely amazing *camera* → good great nice	*phone* → great smart beautiful *reception* → perfect
	−1	*camera* → big crappy defective *lens* → vertical cheap wide	*phone* → worst crappy shittest *battery life* → low
Samsung	+1	*camera* → gorgeous great cool *pics* → nice great perfect	*phone* → mobile great nice *service* → good sweet friendly
	−1	*camera* → digital free crazy *shots* → quick wide	*phone* → stupid bad fake *battery life* → solid poor terrible

In Table 6.11, we display an extract of contrasting tweets containing the target 'iphone' with positive ($r = 1$) or negative ($r = -1$) sentiment. The advantage of the TOTM over the other methods is that the positive and negative opinions are directed toward the targets. For instance, the TOTM correctly identifies the sentiment of iPhone on the following tweet, thanks to the use of the dependency parser: "*Ah, well there you go. The iPhone is dead, long live Android!*"

6.10 Diagnostics

In this section, we perform some diagnostic tests to assess the inference algorithm of the TOTM. In particular, we look at the training log likelihood during learning to make sure the learned model converges. We also inspect the posterior of the hyperparameter b to verify that the gradient ascent algorithm is working properly.

6.10.1 Convergence Analysis of the Collapsed Gibbs Sampler

It is important to assess the convergence of an MCMC algorithm to avoid premature termination of the algorithm. As mentioned, we say an algorithm has converged when the training log likelihood, $p(\mathbf{W}, \mathbf{O} \mid \mathbf{A}, \mathbf{R}, \psi, \phi')$, do not change by more than 0.5 % in ten consecutive iterations. In Table 6.5, we display the training log likelihood for the TOTM trained on the Electronic dataset. The plot shows that the collapsed Gibbs sampler converges within 500 iterations.

Table 6.11: Contrasting opinions on iPhones. For privacy reason we censor the usernames with a placeholder "@user".

Positive Opinion	Negative Opinion
RT @user : the iPhone is so awesome!!! Emailing, texting, surfing the sametime! — Can do all *tgat* while talkin on the phone?...	@user awww thx! I can't send an email right now bc my iPhone is stupid with sending emails. Lol but I can tweet or dm u?
Ahhh! Tweeting on my gorgeous iPhone! I missed you! hehe am on my way home, put the kettle on will you pls :)	It would appear that the iPhone, due to construction, is weak at holding signal. Combine that with a bullshit 3G network in Denver.
Thanks @user for the link to iPhone vs Blackberry debate. I got the iPhone & it's just magic! So intuitive!	@user @user Ah, well there you go. The iPhone is dead, long live Android! ;)
Finally my fave lover @user has Twitter & will be using it all the time with her cool new iPhone :)	@user Finally eh? :D I think iphone is so ugly x.x

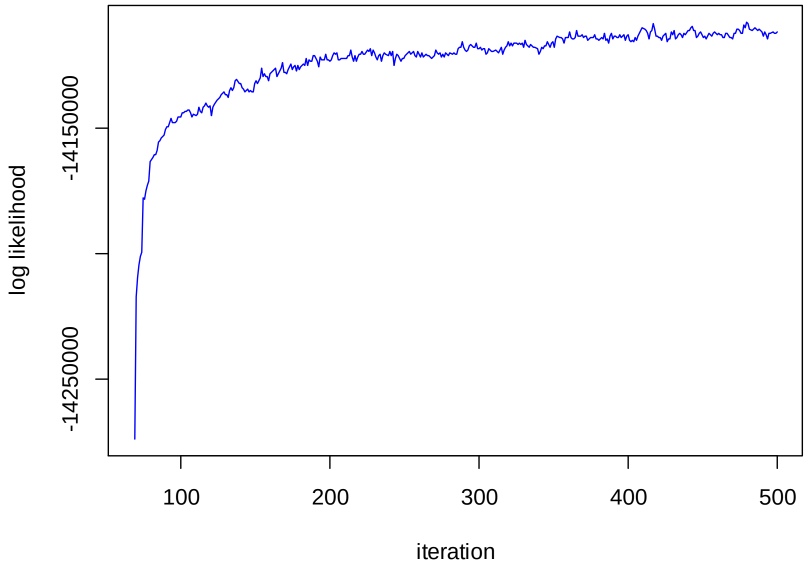


Figure 6.5: Convergence analysis for the TOTM. The plot shows that the training log likelihood $p(\mathbf{W}, \mathbf{O} \mid \mathbf{A}, \mathbf{R}, \psi, \phi')$ increases quickly initially, and then converges.

6.10.2 Inspecting the Posterior of the Sentiment Hyperparameter

The hyperparameter b was introduced to control the strength of the prior information from the sentiment lexicon. Instead of manually tuning the hyperparameter, we have

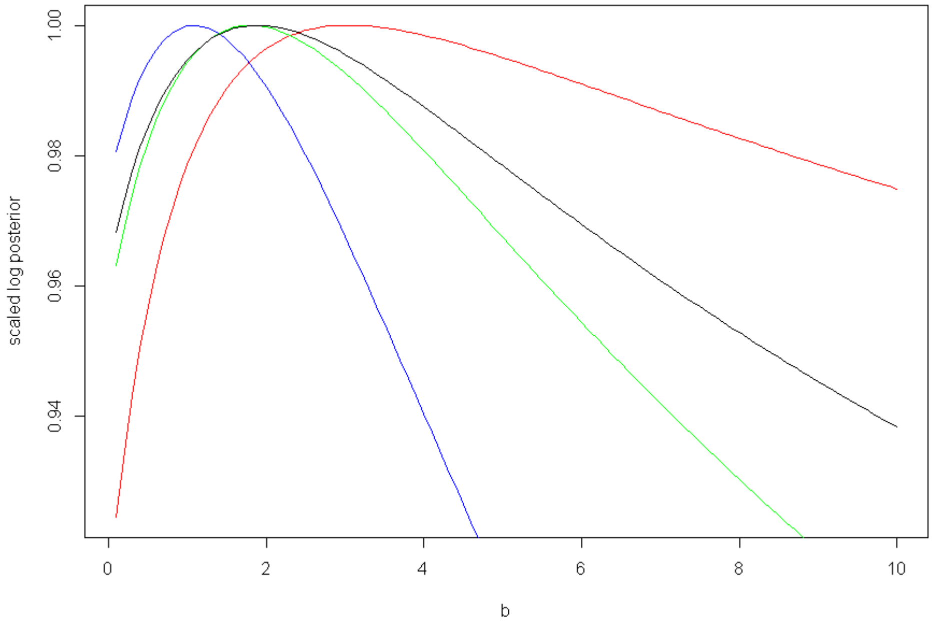


Figure 6.6: The log posteriors of the hyperparameter b corresponds to different values of c_{rv} and X_{rv} during the inference procedure. The log posteriors are scaled so they can be shown in the same plot. The plot shows that the posteriors are unimodal and thus can easily be optimised with a gradient ascent algorithm.

introduced a gradient ascent algorithm to automatically learn the hyperparameter. Here, we look at the log posterior of the hyperparameter b given various statistics c_{rv} from obtained during the inference algorithm. We present the plot in Figure 6.6, the log likelihood curves are scaled such that they all fit in the same plot. From Figure 6.6, we see that the log posterior for b is unimodal. Thus, we are assured that the gradient ascent algorithm will be able to find the optimal b.

6.11 Summary

In this chapter, we study the use of a nonparametric Bayesian topic model for opinion analysis on tweets, focussing on a tweet corpus queried with electronic product terms. This is motivated by the fact that Twitter is a popular platform for opinions and that tweets are publicly available. Unlike reviews, tweets do not contain scores or ratings, they are more informal and usually accompanied by emoticons and strong sentiment words. Taking advantage of the informal nature of tweets, we design a topic model named TOTM for opinion analysis. The TOTM is shown to greatly improve opinion prediction with a direct target–opinion modelling. In incorporating a sentiment lexicon into topic models, we propose a new formulation for the topic model priors, which learns and updates itself given data. Our innovative formulation is shown to improve sentiment analysis significantly.

Our qualitative analysis demonstrates that opinion mining on tweets provide useful opinions on electronic products. Note that although we can obtain a large quantity of product opinions on tweets, the opinions are usually much noisier than that of the reviews. For instance, opinions can be incidental (*e.g.*, the user was just frustrated with the product at that time), since it is easy and effortless to produce a tweet. As with the reviews, the opinions on tweets may not always be true. Some tweets are laden with sarcasm, making them difficult to interpret, while some others are spam containing no useful information. Additionally, be aware that the opinions extracted from Twitter do not represent the public opinions since not everyone use Twitter.

We emphasise the importance of the preprocessing steps. For instance, word normalisation allows misspellings and abbreviations to be captured for target–opinion analysis; tweet aggregation improves aspect clustering and lets us compare different products or brands. For practical applications, filtering sarcastic tweets and spam is also important. In this chapter, we have attempted to filter spam by removing tweets containing URLs. We acknowledge that although there is existing work on removing sarcastic tweets and spam [Tsur *et al.*, 2010; McCord and Chuah, 2011], we did not incorporate them due to the lack of publicly available software.

Future work on this area includes the incorporation other word lexicons, such as synonym and antonym lexicons, into topic models for sentiment analysis. In the next chapter, we will move away from sentiment analysis and focus on developing a topic model for bibliographic analysis, particularly on research publications.

Chapter 7

Bibliographic Analysis on Research Publications using Authors and Citation Network

Bibliographic analysis considers the author's research areas, the citation network and the paper content among other things. In this chapter, we combine these three in a topic model that produces a bibliographic model of authors, topics and documents, using a nonparametric extension of a combination of the Poisson mixed-topic link model and the author-topic model. This gives rise to the Citation Network Topic Model (CNTM). We propose a novel and efficient inference algorithm for the CNTM to explore subsets of research publications from CiteSeer$^{\text{X}}$. Our model demonstrates improved performance in both model fitting and a clustering task compared to several baselines. Additionally, we propose a simple method to incorporate supervision into topic modelling to achieve further improvement on the clustering task. This chapter is an extension of Lim and Buntine [2014a].

7.1 Introduction

Models of bibliographic data need to consider many kinds of information. Articles are usually accompanied by metadata, such as authors, publishers, categories, and time. Cited papers can also be available. When authors' topic preferences are modelled, we need to associate the document topic information somehow with the authors. Jointly modelling text data with citation network information can be challenging, and the problem is confounded when also modelling author–topic relationships.

In this chapter, we propose a topic model to jointly model authors' topic preferences, text content[27] and the citation network. The model is a nonparametric extension of previous models discussed in Section 7.2. Using simple assumptions and approximations, we derive a novel algorithm that allows the probability vectors in the model to be integrated out; this gives an MCMC inference *via* discrete sampling. Ad-

[27] Abstract and publication title.

ditionally, we propose a fully supervised approach into topic modelling to improve document clustering, by making use of categorical information that is available. As applications, we employ the CNTM for document clustering and extraction of topical summary from the clusters. In addition, we perform qualitative analysis to further investigate the authors' research areas and visualise the author–topics network.

The rest of this chapter is organised as follows. Sections 7.3, 7.4 and 7.5 detail the CNTM and its inference algorithm. We describe the datasets used in Section 7.6 and report on the experiments in Section 7.7. Applying our model on research publication data, we demonstrate the improved performance of the model, on both model fitting and a clustering task, compared to several baselines. In Section 7.8, we analyse the inference results produced by the CNTM qualitatively. We find that the learned topics have high comprehensibility. Additionally, we present a visualisation snapshot of the learned topic models. Finally, we perform diagnostic assessment of our topic models in Section 7.9 and summarise this chapter in Section 7.10.

7.2 Related Work

As demonstrated by the previous chapter, variants of LDA allow incorporating more aspects of a particular task; here we consider authorship and citation information. The author-topic model (ATM) [Rosen-Zvi *et al.*, 2004] uses the authorship information to restrict topic options based on the author. Some recent work jointly models the document citation network and text content. This includes the *relational topic model* [Chang and Blei, 2010], the *Poisson mixed-topic link model* (PMTLM) [Zhu *et al.*, 2013] and *Link-PLSA-LDA* [Nallapati *et al.*, 2008]. An extensive review of these models can be found in Zhu *et al.* [2013]. The *Citation Author Topic* (CAT) model [Tu *et al.*, 2010] models the author-author network on publications based on citations using an extension of the ATM. Note that our work is different to CAT in that we model the author-document-citation network instead of the author-author network.

The *Topic-Link LDA* [Liu *et al.*, 2009] jointly models author and text by using the distance between the document and author topic vectors. Similarly the Twitter Network topic model that will be discussed in next chapter models the author network[28] based on author topic distributions, but using a Gaussian process to model the network. Note that our work considers the author-document-citation of Liu *et al.* [2009]. We use the PMTLM of Zhu *et al.* [2013] to model the network, which lets one integrate PYP hierarchies with the PMTLM using efficient MCMC sampling.

There is also existing work on analysing the degree of authors' influence. On publication data, Kataria *et al.* [2011] and Mimno and McCallum [2007] analyse influential authors with topic models. While Weng *et al.* [2010], Tang *et al.* [2009] and Liu *et al.* [2010] use topic models to analyse users' influence on social media.

[28] The author network here corresponds to the Twitter follower network.

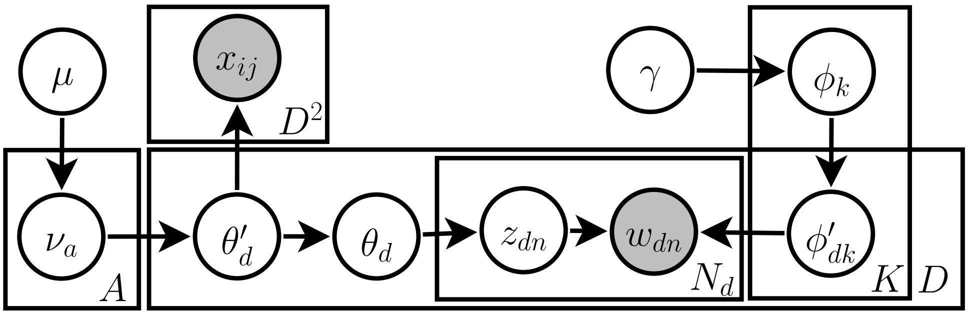


Figure 7.1: Graphical model for the Citation Network Topic Model (CNTM). The shaded nodes represent observed variables while the unshaded nodes are latent. The box on the top left with D^2 entries is the citation network on documents represented as a Boolean matrix. The remainder is a nonparametric author-topic model where the A authors on the left have topic distributions that influence the D document–topic distributions. The K topics, shown in the top right, have bursty modelling following Buntine and Mishra [2014].

7.3 Citation Network Topic Model

In this section, we propose a topic model that jointly model the *text*, *authors*, and the *citation network* of research publications (documents). We name the topic model the Citation Network Topic Model (CNTM). We first describe the topic model part of the CNTM where the citations are not considered, which will be used for comparison later in Section 7.7. We then complete the CNTM with the discussion on its network component. Additionally, we propose a novel approach to incorporate supervision into CNTM, which is detailed in Section 7.3.3. The full graphical model for CNTM is displayed in Figure 7.1.

7.3.1 Hierarchical Pitman-Yor Process Topic Model

In modelling authorship, the CNTM modifies the approach of the ATM, which assumes that the words in a publication are equally attributed to the different authors. This is not reflected in practice since publications are often mainly written by the first author, except when the order is alphabetical. Thus, we assume that the first author is dominant and attribute all the words in a publication to the first author. Although we could model the contribution of each author on a publication by, say, using a Dirichlet distribution, we found that considering only the first author gives a simpler learning algorithm and cleaner results.

The generation process of the topic model component of the CNTM is as follows. We first sample a root topic distribution μ with a GEM distribution to act as a base distribution for the author–topic distributions ν_a for each author a, as follows:

$$\mu \sim \mathrm{GEM}(\alpha^{\mu}, \beta^{\mu}) \, , \tag{7.1}$$

$$\nu_a \,|\, \mu \sim \mathrm{PYP}(\alpha^{\nu_a}, \beta^{\nu_a}, \mu) \, , \qquad \text{for } \; a = 1, \ldots, A \, . \tag{7.2}$$

Given the first author a_d of each publication d, we sample the document–topic prior θ'_d and the document–topic distribution θ_d , as the following:

$$\theta'_d \,|\, a_d, \nu \sim \mathrm{PYP}(\alpha^{\theta'_d}, \beta^{\theta'_d}, \nu_{a_d}) \, , \tag{7.3}$$

$$\theta_d \,|\, \theta'_d \sim \mathrm{PYP}(\alpha^{\theta_d}, \beta^{\theta_d}, \theta'_d) \, , \qquad \text{for } \; d = 1, \ldots, D \, . \tag{7.4}$$

Note that instead of modelling a single document–topic distribution, we model a document–topic hierarchy with θ' and θ. The primed θ' represents the topics of the document in the context of the citation network. The unprimed θ represents the topics of the text, naturally related to θ' but not the same. Such modelling gives citation information a higher impact, taking into account the relatively low amount of citations compared to the text. The technical details on the effect of such modelling is presented in Section 7.9.2.

On the vocabulary side, we generate a background word distribution γ given H^{γ}, a discrete uniform vector of length $|\mathcal{V}|$, that is, $H^{\gamma} = (\ldots, \frac{1}{|\mathcal{V}|}, \ldots)$. The symbol $\mathcal{V}$ denotes the set of *distinct* word tokens observed in a corpus. Then, we sample a topic–word distribution ϕ_k for each topic k, with γ as the base distribution. This is illustrated by

$$\gamma \sim \mathrm{PYP}(\alpha^{\gamma}, \beta^{\gamma}, H^{\gamma}) \, , \tag{7.5}$$

$$\phi_k \,|\, \gamma \sim \mathrm{PYP}(\alpha^{\phi_k}, \beta^{\phi_k}, \gamma) \, , \qquad \text{for } \; k = 1, \ldots, K \, . \tag{7.6}$$

Modelling word burstiness is important since words in a document are likely to repeat in the document [Buntine and Mishra, 2014]. The same applies to publication abstract, as illustrated by Table 7.2 in Section 7.6. To address this property, we make the topics bursty so each document only focusses on a subset of words in the topic. This is achieved by defining the ϕ'_{dk} for each topic k in document d as

$$\phi'_{dk} \,|\, \phi_k \sim \mathrm{PYP}(\alpha^{\phi'_{dk}}, \beta^{\phi'_{dk}}, \phi_k) \, . \tag{7.7}$$

Finally, for each word w_{dn} in document d, we sample the corresponding topic assignment z_{dn} from the document–topic distribution θ_d; while the word w_{dn} is sampled from the topic–word distribution ϕ'_d given z_{dn} , that is,

$$z_{dn} \,|\, \theta_d \sim \mathrm{Discrete}(\theta_d) \, , \tag{7.8}$$

$$w_{dn} \,|\, z_{dn}, \phi'_d \sim \mathrm{Discrete}(\phi'_{dz_{dn}}) \, , \qquad \text{for } \; n = 1, \ldots, N_d \, . \tag{7.9}$$

Note that w includes words from the title and the abstract of the publications, but not the full article. This is because title and abstract provide a good summary of the topics of a given publication and thus more suited for topic modelling, while the full article might contain too much technical details that are not too relevant.

In the next section, we carry through to the modelling of the citation network accompanying a publication collections. This completes the CNTM.

7.3.2 Citation Network Poisson Model

To model the citation network between publications, we assume that the citations are generated conditioned on the topic distributions of the publications. Our approach is motivated by the degree-corrected variant of PMTLM [Zhu *et al.*, 2013]. Denoting x_{ij} as the number of times document i citing document j, we model x_{ij} with a Poisson distribution with mean parameter λ_{ij}, namely,

$$x_{ij} \,|\, \lambda_{ij} \sim \text{Poisson}(\lambda_{ij}), \qquad \text{for } \; i = 1, \ldots, D; \;\; j = 1, \ldots, D, \tag{7.10}$$

where $\lambda_{ij} = \lambda_i^+ \lambda_j^- \sum_k \lambda_k^T \theta'_{ik} \theta'_{jk}$.

Here, λ_i^+ is the propensity of document i to cite and λ_j^- represents the popularity of cited document j. The parameter λ_k^T scales the k-th topic, effectively penalising common topics and strengthen rare topics. Hence, a citation from document i to document j is more likely when these documents are having relevant topics. The Poisson distribution is used instead of a Bernoulli distribution because it leads to dramatically reduced complexity in analysis [Zhu *et al.*, 2013]. We note that the Poisson distribution is similar to the Bernoulli distribution when the mean parameter is small. We present a list of variables associated with the CNTM in Table 7.1.

7.3.3 Incorporating Supervision into CNTM

Author modelling allows topic sharing of multiple documents written by the same author. However, there are many authors who have authored only a few publications. Their treatment can be problematic to topic models that incorporate authorship information since these authors are not discriminative enough for prediction. We propose an extension of the CNTM to address this issue. We name this extension the supervised CNTM (SCNTM) as it allows supervision during model training.

We introduce author threshold η, a parameter that controls the level of supervision used by the SCNTM. In the SCNTM, for each document, if the author has produced less than η publications, the authorship is replaced by a *dummy author* that corresponds to the categorical label of the document. These authors are effectively merged to form a collection of authors with similar research area. For example, $\eta = 2$ means authors who have only a *single* publication are replaced, while $\eta = 1$ corresponds to no replacement. In addition to being able to cluster the documents

Table 7.1: List of Variables for the Citation Network Topic Model (CNTM).

Variable	Name	Description
z_{dn}	Topic	Category (topical) label for word w_{dn}.
w_{dn}	Word	Observed word or phrase at position n in document d.
x_{ij}	Citations	Number of times document i cites document j.
ϕ'_{dk}	Word distribution	Probability distribution in generating words given document d and topic k.
ϕ_k	Topic–word distribution	Word prior for ϕ'_{dk}.
θ_d	Document–topic distribution	Probability distribution in generating topics for document d.
θ'_d	Document–topic prior	Topic prior for θ_d.
ν_a	Author–topic distribution	Probability distribution in generating topics for author a.
γ	Global word distribution	Word prior for ϕ_k.
μ	Global topic distribution	Topic prior for ν_a.
$\alpha^{\mathcal{N}}$	Discount	Discount parameter for PYP $\mathcal{N}$.
$\beta^{\mathcal{N}}$	Concentration	Concentration parameter for PYP $\mathcal{N}$.
$H^{\mathcal{N}}$	Base distribution	Base distribution of the PYP $\mathcal{N}$.
λ_{ij}	Rate	Rate parameter or the mean for x_{ij}.
λ_i^+	Cite propensity	Propensity to cite for document i.
λ_i^-	Cited propensity	Propensity to be cited for document j.
λ_k^T	Scaling factor	Citation scaling factor for topic k.

better, as shown in Section 7.7.4, the SCNTM achieves a reduction of memory usage by greatly reducing the number of authors that need to be modelled.

7.4 Model Likelihood

In this section, we present the posterior likelihood of the CNTM. Note that we have used the CRP representation as discussed in Section 5.3. As before, we will use c to denote customer counts and t to denote table counts.

7.4.1 Posterior Likelihood for the HPYP Topic Model

Deriving the posterior likelihood for the topic model part of the CNTM follows the reasoning in Section 5.3, so we will keep it brief. As before, we use bold face capital letters to denote the set of all relevant lower case variables. For example, $\mathbf{Z} = \{z_{11}, \cdots, z_{DN_D}\}$ denotes the set of all topic assignments. Additionally, we denote Ξ as the set of all hyperparameters, whether they are from the HPYP topic model (α, β), or from the citation network Poisson model (λ). With the CRP representation, we can write the posterior of the HPYP topic model in a modularised form, as follows:

$$\begin{aligned} p(\mathbf{Z}, \mathbf{T}, \mathbf{C} \,|\, \mathbf{W}, \Xi) &\propto p(\mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C} \,|\, \Xi) \\ &\propto f(\mu) \left(\prod_{a=1}^{A} f(\nu_a) \right) \left(\prod_{d=1}^{D} f(\theta'_d)\, f(\theta_d) \prod_{k=1}^{K} f(\phi'_{dk}) \right) \left(\prod_{k=1}^{K} f(\phi_k) \right) \\ &\qquad f(\gamma) \left(\prod_{v=1}^{|\mathcal{V}|} \left(\frac{1}{|\mathcal{V}|} \right)^{t_v^{\gamma}} \right), \end{aligned} \tag{7.11}$$

as before, the function $f(\mathcal{N})$ is the modularised likelihood for the variable $\mathcal{N}$ defined in Equation (5.9).

7.4.2 Posterior Likelihood for the Citation Network Poisson Model

For the citation network, the Poisson likelihood for each x_{ij} is given as

$$p(x_{ij} \,|\, \lambda, \theta') = \frac{\lambda_{ij}^{x_{ij}}}{x_{ij}!\, e^{\lambda_{ij}}} = \left(\lambda_i^+ \lambda_j^- \sum_{k=1}^{K} \lambda_k^T \theta'_{ik} \theta'_{jk} \right)^{x_{ij}} \exp\left(-\lambda_i^+ \lambda_j^- \sum_{k=1}^{K} \lambda_k^T \theta'_{ik} \theta'_{jk} \right). \tag{7.12}$$

Note that the term $x_{ij}!$ is dropped in Equation (7.12) since x_{ij} can only be 0 or 1 given the format of the data thus $x_{ij}!$ is evaluated to 1. With conditional independence of the x_{ij} given θ', the joint likelihood for the whole citation network $\mathbf{X} = \{x_{11}, \cdots, x_{DD}\}$ can be written as

$$\begin{aligned} p(\mathbf{X} \,|\, \lambda, \theta') = &\left(\prod_{i=1}^{D} (\lambda_i^+)^{g_i^+} \, (\lambda_i^-)^{g_i^-} \right) \left[\prod_{i=1}^{D} \prod_{j=1}^{D} \left(\sum_{k=1}^{K} \lambda_k^T \theta'_{ik} \theta'_{jk} \right)^{x_{ij}} \right] \\ &\exp\left(- \sum_{i=1}^{D} \sum_{j=1}^{D} \sum_{k=1}^{K} \lambda_i^+ \lambda_j^- \lambda_k^T \theta'_{ik} \theta'_{jk} \right), \end{aligned} \tag{7.13}$$

where g_i^+ is the number of citations in publication i, that is, $g_i^+ = \sum_j x_{ij}$, and g_i^- is the number of times publication i being cited, that is, $g_i^- = \sum_j x_{ji}$. We also make a simplifying assumption that $x_{ii} = 1$ for all documents, that is, all publications are treated as self-cited. This assumption is important since defining x_{ii} allows us to rewrite the joint likelihood into Equation (7.13), which is crucial for efficient caching during inference. This reduces the polynomial time complexity into a linear one, see Section 7.9.3 for details. The full posterior of the CNTM is simply the product of the two posteriors from Equation (7.11) and Equation (7.13), namely,

$$p(\mathbf{X},\mathbf{Z},\mathbf{T},\mathbf{C}\,|\,\mathbf{W},\mathbf{X}) = p(\mathbf{Z},\mathbf{T},\mathbf{C}\,|\,\mathbf{W},\Xi)\,p\big(\mathbf{X}\,|\,\lambda,\theta'\big). \tag{7.14}$$

In the next section, we demonstrate that our model representation gives rise to an intuitive sampling algorithm for learning the model. We also show how the Poisson model integrates into the topic modelling framework.

7.5 Inference Techniques

Here, we derive the MCMC algorithms for training the CNTM.[29] We first discuss the collapsed Gibbs sampler for the HPYP topic model and then describe the Metropolis-Hastings (MH) algorithm for the citation network. The full inference procedure is performed by alternating between the collapsed Gibbs sampler and the MH algorithm. Finally, we outline the hyperparameters samplers.

7.5.1 Collapsed Gibbs Sampler for the HPYP Topic Model

We adopt the collapsed Gibbs sampler developed in Section 5.3 for the learning of the HPYP topic model. The only difference for the HPYP topic model in this chapter is the formulation of the joint conditional posterior distribution.

To illustrate, the joint conditional posterior distribution used in the blocked Gibbs sampler is given as

$$p\big(z_{dn},\mathbf{T},\mathbf{C}\,|\,\mathbf{Z}^{-dn},\mathbf{W},\mathbf{T}^{-dn},\mathbf{C}^{-dn},\Xi\big) = \frac{p(\mathbf{Z},\mathbf{T},\mathbf{C}\,|\,\mathbf{W},\Xi)}{p(\mathbf{Z}^{-dn},\mathbf{T}^{-dn},\mathbf{C}^{-dn}\,|\,\mathbf{W},\Xi)}\,. \tag{7.15}$$

As before, the superscript $\square^{-dn}$ indicates that the word w_{dn} and its associated variables are removed from the respective sets.

7.5.2 Metropolis-Hastings Algorithm for the Citation Network

A naïve MH algorithm can be proposed for learning the citation network. For example, to sample the document topic distribution θ'_d given $\mathbf{X}$, we can propose a new

[29] Note that SCNTM follows the same inference algorithm as CNTM.

θ'_d with a Dirichlet distribution given the parent vector ν_{a_d}, and accept or reject the proposal following an MH scheme. However, this algorithm requires the probability vectors (θ', ν, μ) to be stored explicitly (as probability vectors, rather than counts) and subsequently the collapsed Gibbs sampler in Section 7.5.1 would be considerably more complicated.

Instead, we propose a novel MH algorithm that allows the probability vectors to remain integrated out, thus retaining the fast discrete sampling procedure for the PYP hierarchy, rather than, for instance, resorting to an expectation-maximisation (EM) algorithm or variational approach. We introduce an auxiliary variable y_{ij}, named *citing topic*, to denote the topic that prompts publication i to cite publication j. To illustrate, for a *biology* publication that cites a *machine learning* publication for the learning technique, the citing topic would be 'machine learning' instead of 'biology'. From Equation (7.10), we model the citing topic y_{ij} as jointly Poisson with x_{ij}, as follows:

$$x_{ij}, y_{ij} = k \mid \lambda, \theta' \sim \text{Poisson}\big(\lambda_i^+ \lambda_j^- \lambda_k^T \theta'_{ik} \theta'_{jk}\big) . \tag{7.16}$$

Incorporating $\mathbf{Y}$, the set of all y_{ij}, we rewrite the citation network likelihood as

$$\begin{aligned} p(\mathbf{X}, \mathbf{Y} \mid \lambda, \theta') \propto &\left(\prod_{i=1}^{D} (\lambda_i^+)^{g_i^+} (\lambda_i^-)^{g_i^-} \right) \left(\prod_{k=1}^{K} (\lambda_k^T)^{\frac{1}{2} \sum_i h_{ik}} \right) \left(\prod_{i=1}^{D} \prod_{k=1}^{K} {\theta'_{ik}}^{h_{ik}} \right) \\ &\exp\left(- \sum_{i=1}^{D} \sum_{j=1}^{D} \lambda_i^+ \lambda_j^- \lambda_{y_{ij}}^T \theta'_{i y_{ij}} \theta'_{j y_{ij}} \right) , \end{aligned} \tag{7.17}$$

where $h_{ik} = \sum_j x_{ij} I(y_{ij} = k) + \sum_j x_{ji} I(y_{ji} = k)$ is the number of connections publication i made due to topic k. Note that $I(\cdot)$ denotes the indicator function. We further note that we can only rewrite the likelihood into this form after defining x_{ii} as in Section 7.4.2.

To integrate out θ', we note the term ${\theta'_{ik}}^{h_{ik}}$ appears like a multinomial likelihood, so we absorb them into the likelihood for $p(\mathbf{Z}, \mathbf{T}, \mathbf{C} \mid \mathbf{W}, \Xi)$ where they correspond to additional counts for $c^{\theta'_i}$, with h_{ik} added to $c_k^{\theta'_i}$. To disambiguate the source of the counts, we will refer these customer counts contributed by x_{ij} as *network counts*, and denote the augmented counts as $\mathbf{C}^+$ ($\mathbf{C}$ plus network counts). For the exponential term, we use the delta method approximation [Oehlert, 1992],

$$\int f(\theta) \, \exp\big(- g(\theta) \big) \, \mathrm{d}\theta \approx \exp\Big(- g(\hat{\theta}) \Big) \int f(\theta) \, \mathrm{d}\theta , \tag{7.18}$$

where $\hat{\theta}$ is the expected value according to a distribution proportional to $f(\theta)$. This approximation is reasonable as long as the terms in the exponential are small (see

Appendix A.2). The approximate full posterior of the CNTM can then be written as

$$p(\mathbf{Z},\mathbf{W},\mathbf{T},\mathbf{C},\mathbf{X},\mathbf{Y}|\lambda,\Xi) \approx p(\mathbf{Z},\mathbf{W},\mathbf{T},\mathbf{C}^{+}|\Xi)\left(\prod_{i=1}^{D}(\lambda_i^{+})^{g_i^{+}}(\lambda_i^{-})^{g_i^{-}}\right)\left(\prod_{k=1}^{K}(\lambda_k^{T})^{\frac{1}{2}\sum_i h_{ik}}\right)$$
$$\exp\left(-\sum_{i=1}^{D}\sum_{j=1}^{D}\lambda_i^{+}\lambda_j^{-}\lambda_{y_{ij}}^{T}\hat{\theta}'_{iy_{ij}}\hat{\theta}'_{jy_{ij}}\right). \tag{7.19}$$

The MH algorithm can be summarised in three steps: estimate the document topic prior θ', propose a new citing topic y_{ij}, and accept or reject the proposed y_{ij} following an MH scheme. Note that the MH algorithm is similar to the collapsed Gibbs sampler, where we decrement the counts, sample a new state and update the counts. Since all probability vectors are represented as counts, we do not need to deal with their vector form in the collapsed Gibbs sampler. Additionally, our MH algorithm is intuitive and simple to implement. Like the words in a document, each citation is assigned a topic, hence the words and citations can be thought as voting to determine the topic of the documents.

The detail for the MH algorithm for the citation network is as follows. First, for each document d, we estimate the expected document–topic prior $\hat{\theta}'_d$

$$\hat{\theta}'_d = \left(\cdots, \frac{(\alpha^{\theta'_d}T^{\theta'_d} + \beta^{\theta'_d})\hat{\nu}_{a_d k} + c_k^{\theta'_d} - \alpha^{\theta'_d}T_k^{\theta'_d}}{\beta^{\theta'_d} + C^{\theta'_d}}, \cdots\right), \tag{7.20}$$

where $\hat{\nu}_{a_i}$ in Equation (7.20) is recursively computed from $\hat{\mu}$ and its associated counts, see Section 5.4.4 for details.

Then, for each document pair (i,j) that satisfies $x_{ij} = 1$, we decrement the network counts associated with x_{ij}, and re-sample y_{ij} with the proposal distribution derived from Equation (7.16):

$$p\left(y_{ij}^{\text{new}} = k \,\middle|\, \hat{\theta}'_i, \hat{\theta}'_j\right) \propto \lambda_k^T\,\hat{\theta}'_{ik}\,\hat{\theta}'_{jk}\,\exp\left(-\lambda_i^{+}\,\lambda_j^{-}\,\lambda_k^T\,\hat{\theta}'_{ik}\,\hat{\theta}'_{jk}\right), \tag{7.21}$$

which can be further simplified since the terms inside the exponential are very small, hence the exponential term approximates to 1. We empirically inspected the exponential term and we found that almost all of them are between 0.99 and 1. This means the ratio of the exponentials is not significant for sampling a new citing topic y_{ij}^{new}. So we ignore the exponential term and let

$$p\left(y_{ij}^{\text{new}} = k \,\middle|\, \hat{\theta}'_i, \hat{\theta}'_j\right) \propto \lambda_k^T\,\hat{\theta}'_{ik}\,\hat{\theta}'_{jk}\,. \tag{7.22}$$

We use the superscripts $\square^{\text{new}}$ and $\square^{\text{old}}$ to denote the proposed sample and the old value respectively. We compute the acceptance probability $A' = \min(A, 1)$ for the newly sampled $y_{ij}^{\text{new}} = y'$, changed from $y_{ij}^{\text{old}} = y^*$, and the successive change to the document–topic priors from $\hat{\theta}'^{\text{old}}$ to $\hat{\theta}'^{\text{new}}$. In the following, we abuse the notations i and j, where the i and j in the summation indexes all documents instead of pointing to particular document i and document j. We decided against introducing additional variables to avoid making the equation more confusing. The acceptance ratio A is given as follows:

$$\begin{aligned} A &= \frac{\exp\Big(-\sum_{ijk} \lambda_i^+ \lambda_j^- \lambda_k^T \, \hat{\theta}'^{\text{new}}_{ik} \, \hat{\theta}'^{\text{new}}_{jk}\Big)}{\exp\Big(-\sum_{ijk} \lambda_i^+ \lambda_j^- \lambda_k^T \, \hat{\theta}'^{\text{old}}_{ik} \, \hat{\theta}'^{\text{old}}_{jk}\Big)} \frac{p\big(\mathbf{Z}, \mathbf{T}, \mathbf{C}^{+\text{new}} \,\big|\, \mathbf{W}, \Xi\big)}{p\big(\mathbf{Z}, \mathbf{T}, \mathbf{C}^{+\text{old}} \,\big|\, \mathbf{W}, \Xi\big)} \\ &\quad \times \frac{\lambda_{y^*}^T \, \hat{\theta}'^{\text{new}}_{iy^*} \, \hat{\theta}'^{\text{new}}_{jy^*}}{\lambda_{y'}^T \, \hat{\theta}'^{\text{old}}_{iy'} \, \hat{\theta}'^{\text{old}}_{jy'}} \frac{\sum_k \lambda_k^T \, \hat{\theta}'^{\text{old}}_{ik} \, \hat{\theta}'^{\text{old}}_{jk}}{\sum_k \lambda_k^T \, \hat{\theta}'^{\text{new}}_{ik} \, \hat{\theta}'^{\text{new}}_{jk}} \,. \end{aligned} \tag{7.23}$$

Finally, if the sample is accepted, we update y_{ij} and the associated customer counts. Otherwise, we discard the sample and revert the changes.

7.5.3 Hyperparameter Sampling

As mentioned in Section 5.4.3, hyperparameter sampling for the priors are important. In our inference algorithm, we sample the concentration parameters β of all PYPs with an auxiliary variable sampler [Teh, 2006a], but leave the discount parameters α fixed. We do not sample the α due to the coupling of the parameter with the Stirling numbers cache. The detail in sampling the hyperparameter β is discussed in Section 5.4.3 and thus not covered here.

In addition to the PYP hyperparameters, we also sample λ^+, λ^- and λ^T with a Gibbs sampler. We let the hyperpriors for λ^+, λ^- and λ^T to be gamma distributed with shape ϵ_0 and rate ϵ_1, that is,

$$\lambda_i^+ \sim \text{Gamma}(\epsilon_0, \epsilon_1), \tag{7.24}$$

$$\lambda_i^- \sim \text{Gamma}(\epsilon_0, \epsilon_1), \tag{7.25}$$

$$\lambda_k^T \sim \text{Gamma}(\epsilon_0, \epsilon_1). \tag{7.26}$$

With the conjugate gamma prior, the posteriors for λ_i^+, λ_i^- and λ_k^T are also gamma distributed, so they can be sampled directly.

$$\lambda_i^+ \,|\, \mathbf{X}, \lambda^-, \lambda^T \theta' \sim \text{Gamma}\Big(\epsilon_0 + g_i^+, \, \epsilon_1 + \textstyle\sum_k \lambda_k^T \theta'_{ik} \sum_j \lambda_j^- \theta'_{jk}\Big), \tag{7.27}$$

$$\lambda_i^- \,|\, \mathbf{X}, \lambda^+, \lambda^T \theta' \sim \text{Gamma}\Big(\epsilon_0 + g_i^-, \, \epsilon_1 + \textstyle\sum_k \lambda_k^T \theta'_{ik} \sum_j \lambda_j^+ \theta'_{jk}\Big), \tag{7.28}$$

$$\lambda_k^T \,|\, \mathbf{X}, \mathbf{Y}, \lambda^+, \lambda^-, \theta' \sim \text{Gamma}\Big(\epsilon_0 + \tfrac{1}{2} \textstyle\sum_i h_{ik}, \, \epsilon_1 + \lambda_k^T \big(\sum_j \lambda_j^+ \theta'_{jk}\big) \big(\sum_j \lambda_j^- \theta'_{jk}\big)\Big). \tag{7.29}$$

We apply vague priors to the hyperpriors by setting $\epsilon_0 = \epsilon_1 = 1$.

Algorithm 7.1 Inference Algorithm for the CNTM

```
1. Initialise the model by assigning a random topic assignment $z_{dn}$ to each word
   $w_{dn}$ and constructing the relevant customer counts $c_k^{\mathcal{N}}$ and table counts $t_k^{\mathcal{N}}$ for
   all variables $\mathcal{N}$.
2. For each word $w_{dn}$ in each document $d$, perform the following:
   (a) Decrement the counts associated with $z_{dn}$ and $w_{dn}$ .
   (b) Blocked sample a new topic $z_{dn}$ and the associated counts $\mathbf{T}$ and $\mathbf{C}$ with
       Equation (7.15).
3. For each citation $x_{ij}$ , perform the following:
   (a) Decrement the network counts associated with $x_{ij}$ and $y_{ij}$ .
   (b) Sample a new citing topic $y_{ij}$ from the joint posterior given by Equa-
       tion (7.22).
   (c) Accept or reject the sampled $y_{ij}$ with an MH scheme with acceptance prob-
       ability given by Equation (7.23).
4. Update the hyperparameters $\beta$, $\lambda^+$, $\lambda^-$ and $\lambda^T$.
5. Repeat Steps 2–4 until the model converges or when a fix number of iterations
   is reached.
```

The full inference algorithm of the CNTM is the combination of the collapsed Gibbs sampler, the MH algorithm and the hyperparameters sampler. The full inference algorithm is summarised in Algorithm 7.1.

7.6 Data

We perform our experiments on subsets of CiteSeer^X data[30] which consists of scientific publications. Each publication from CiteSeer^X is accompanied by *title*, *abstract*, *keywords*, *authors*, *citations* and other metadata. We prepare three publication datasets from CiteSeer^X for evaluations. The first dataset corresponds to Machine Learning (ML) publications, which are queried from CiteSeer^X using the keywords from Microsoft Academic Search.[31] The ML dataset contains 139,227 publications. Our second dataset corresponds to publications from ten distinct research areas. The query words for these ten disciplines are chosen such that the publications form distinct clusters. We name this dataset M10 (Multidisciplinary 10 classes), which is made of 10,310 publications. For the third dataset, we query publications from both arts and

[30] http://citeseerx.ist.psu.edu/ (last accessed 18 August 2014)

[31] http://academic.research.microsoft.com/ (last accessed 18 August 2014)

science disciplines. Arts publications are made of *history* and *religion* publications, while the science publications contain *physics*, *chemistry* and *biology* researches. This dataset consists of 18,720 publications and is named AvS (Arts versus Science) in this chapter. These queried datasets are made available online.[32]

The keywords used to create the datasets are obtained from Microsoft Academic Search, and are listed in Appendix A.3. For the clustering evaluation in Section 7.7.4, we treat the query categories as the ground truth. However, publications that span multiple disciplines can be problematic for clustering evaluation, hence we simply remove the publications that satisfy the queries from more than one discipline. Nonetheless, the labels are inherently noisy. The metadata for the publications can also be noisy, for instance, the *authors* field may sometimes display the publication keywords instead of the authors, publication title is sometimes an URL, and table of contents can be mistakenly parsed as the abstract. We discuss our treatments to these issues in Section 7.6.1. We also note that non-English publications are discarded using *langid.py* [Lui and Baldwin, 2012].

In addition to the manually queried datasets, we also use the existing datasets from LINQS [Sen *et al.*, 2008][33] to facilitate comparison with existing work. In particular, we use their CiteSeer, Cora and PubMed datasets. Their CiteSeer data consists of Computer Science publications and hence we name the dataset CS to remove ambiguity. Although these datasets are small, they are fully labelled and thus useful for clustering evaluation. However, these three datasets do not come with additional metadata such as the authorship information. Note that the CS and Cora datasets are presented as Boolean matrices, that is, the word counts information is lost and we assume that all words in a document occur only once. Additionally, the words have been converted to integer so they do not convey any semantic. Although this representation is less useful for topic modelling, we still use them for the sake of comparison. For the PubMed dataset, we recover the word counts from TF-IDF using a simple assumption (see Appendix A.4). We present a summary of the datasets in Table 7.2 and their respective categorical labels in Table 7.3.

7.6.1 Removing Noise

Here, we briefly discuss the steps taken in cleansing the noise from the CiteSeer^X datasets (ML, M10 and AvS). Note that the *keywords* field in the publications are often empty and are sometimes noisy, that is, they contain irrelevant information such as section heading and title, which makes the keywords unreliable source of information as categories. Instead, we simply treat the keywords as part of the abstracts. We also remove the URLs from the data since they do not provide any additional useful information.

[32] http://karwai.weebly.com/publications.html (last accessed 18 August 2014)

[33] http://linqs.cs.umd.edu/projects/projects/lbc/ (last accessed 18 August 2014)

Table 7.2: Summary of the datasets used in this chapter, showing the number of publications, citations, authors, unique word tokens, the average number of words in each document, and the average percentage of unique words repeated in a document. Note: author information is not available in the last three datasets.

Dataset	Publications	Citations	Authors	Vocab	Words/Doc	Repeat [%]
ML	139 227	1 105 462	43 643	8 322	59.4	23.3
M10	10 310	77 222	6 423	2 956	57.8	24.3
AvS	18 720	54 601	11 898	4 770	58.9	17.0
CS	3 312	4 608	–	3 703	31.8	–
Cora	2 708	5 429	–	1 433	18.2	–
PubMed	19 717	44 335	–	4 209	67.6	40.1

Table 7.3: Categorical labels of the datasets.

Dataset	Classes	Categorical Labels
ML	1	Machine Learning
M10	10	Agriculture, Archaeology, Biology, Computer Science, Physics, Financial Economics, Industrial Engineering, Material Science, Petroleum Chemistry, Social Science
AvS	5	History, Religion, Physics, Chemistry, Biology
CS	6	Agents, AI, DB, IR, ML, HCI
Cora	7	Case Based, Genetic Algorithms, Neural Networks, Theory, Probabilistic Methods, Reinforcement Learning, Rule Learning
PubMed	3	"Diabetes Mellitus, Experimental", Diabetes Mellitus Type 1, Diabetes Mellitus Type 2

Moreover, the author information is not consistently presented in the CiteSeer^X data. Some of the authors are shown with full name, some with first name initialised, while some others are prefixed with title (Prof, Dr., *etc.*). We thus standardise the author information by removing all title from the authors, initialising all first names and discarding the middle names. Although standardisation allows us to match up the authors, it does not solve the problem that different authors who have the same initial and last name are treated as a single author. For example, both Bruce Lee and Brett Lee are standardised to B Lee. Note this corresponds to a whole research problem [Han *et al.*, 2004, 2005] and hence not addressed in this dissertation. Occasionally, institutions are mistakenly treated as authors in CiteSeer^X data. Example includes

American Mathematical Society and *Technische Universität München*. In this case, we remove the invalid authors using a list of exclusion words. The list of exclusion words is presented in Appendix A.5.

7.6.2 Text Preprocessing

Here, we discuss the preprocessing pipeline adopted for the *queried* datasets (note LINQS data were already processed). First, since publication text contains many technical terms that are made of multiple words, we tokenise the text using phrases (or collocations) instead of *unigram* words. Thus, phrases like *decision tree* are treated as single token rather than two distinct words. The phrases are extracted from the respective datasets using LingPipe [Carpenter, 2004].[34] As in Chapter 6, we use the word *words* to mean both unigram words and phrases.

We then change all the words to lower case and filter out certain words. Words that are removed are *stop words*, common words and rare words. Note that we use the stop words list from MALLET [McCallum, 2002], we define common words as words that appear in more than 18 % of the publications, and rare words are words that occur less than 50 times in each dataset. Note that the thresholds are determined by inspecting the words removed. Finally, the tokenised words are stored as arrays of integers. We also split the datasets to 90 % training set for training the topic models, and 10 % test set for evaluations detailed in Section 7.7.

7.7 Experiments and Results

In this section, we describe experiments that compare the CNTM and SCNTM against several baseline topic models. The baselines are HDP-LDA with burstiness [Buntine and Mishra, 2014], a nonparametric extension of the ATM, and the PMTLM [Zhu *et al.*, 2013]. We also display the results for the CNTM without the citation network for comparison purpose. We evaluate these models quantitatively with goodness-of-fit and clustering measures.

7.7.1 Experiment Settings

In the following experiments, we initialise the concentration parameters β of all PYPs to 0.1, noting that the hyperparameters are updated automatically. We set the discount parameters α to 0.7 for all PYPs corresponding to the "*word*" side of the CNTM (*i.e.*, γ, ϕ, ϕ'). This is to induce power-law behaviour on the word distributions. We simply set the α to 0.01 for all other PYPs. Note that the number of topics grow with data in nonparametric topic modelling. To prevent the learned topics to be too fine-grained, we set a limit to the maximum number of topics that can be learned. In

[34] http://alias-i.com/lingpipe/ (last accessed 18 August 2014)

particular, we set the number of topics cap to 20 for the ML dataset, 50 for the M10 dataset and 30 for the AvS dataset. For all the topic models, our experiments find that the number of topics always converges to the cap. For CS, Cora and PubMed datasets, we *fix* the number of topics to 6, 7 and 3 respectively for comparison against the PMTLM.

When training the topic models, we run the inference algorithm for 2,000 iterations. For the CNTM, the MH algorithm for the citation network is performed after the 1,000th iteration, this is so the topics can be learned from the collapsed Gibbs sampler first. This gives a faster learning algorithm and also allows us to assess the *"value-added"* by the citation network to topic modelling (see Section 7.9.1). We repeat each experiment five times to reduce the estimation error of the evaluation measures.

7.7.2 Estimating the Test Documents' Topic Distributions

The topic distribution θ' on the test documents are required in performing various evaluations on topic models. These topic distributions are unknown and hence need to be estimated. Standard practice uses the first half of the text in each test document to estimate θ', and uses the other half for evaluations. However, since abstracts are relatively shorter compared to articles, adopting such practice would mean there are too little text to be used for evaluations. Instead, we used only the words from the publication title to estimate θ', allowing more words for evaluation. Moreover, title is also a good indicator of topic so it is well suited to be used in estimating θ'. The estimated θ' will be used in perplexity and clustering evaluations below. We note that for the clustering task, both title and abstract text are used in estimating θ' as there is no need to use the text for clustering evaluation.

We briefly describe how we estimate the topic distributions θ' of the test documents. Denoting w_{dn} to represent the word at position n in a test document d, we estimate the topic assignment z_{dn} of word w_{dn} *independently* by sampling from their predictive posterior distribution given the learned author–topic distributions ν and topic–word distributions ϕ:

$$p(z_{dn} \,|\, w_{dn}, \nu, \phi) \propto \nu_{a_d k} \, \phi_{k w_{dn}} \,, \tag{7.30}$$

noting that the intermediate distributions ϕ' are integrated out (see Appendix A.6).

We then build the customer counts c^{θ_d} for these test documents from the sampled z (for simplicity, we set the corresponding table counts as half the customer counts). With these, we then estimate the document–topic distribution θ' from Equation (7.20). If citation network information is present, we refine the document–topic distribution θ'_d using the linking topic y_{dj} for train document j where $x_{dj} = 1$. The linking topic y_{dj} is sampled from the estimated θ'_d and is added to the customer counts $c^{\theta'_d}$, which further updates the document–topic distribution θ'_d.

Doing the above gives a sample of the document–topic distribution $\theta_d'^{(r)}$. We adopt a Monte Carlo approach by generating $R = 500$ samples of $\theta_d'^{(r)}$, and calculate the Monte Carlo estimate of θ_d', as described in Section 5.5.1.

7.7.3 Goodness-of-fit Test

Perplexity, as detailed in Section 5.5.2, is a popular metric used to evaluate the goodness-of-fit of a topic model. Since perplexity is negatively related to the likelihood of the observed words $\mathbf{W}$ given the model, the lower the better. Here, the perplexity can be computed as:

$$\text{perplexity}(\mathbf{W}) = \exp\left(- \frac{\sum_{d=1}^{D} \sum_{n=1}^{N_d} \log p\big(w_{dn} \,|\, \theta_d', \phi\big)}{\sum_{d=1}^{D} N_d} \right), \tag{7.31}$$

where $p\big(w_{dn} \,|\, \theta_d', \phi\big)$ is obtained by summing over all possible topics:

$$p\big(w_{dn} \,|\, \theta_d', \phi\big) = \sum_{k=1}^{K} p\big(w_{dn} \,|\, z_{dn} = k, \phi_k\big)\, p\big(z_{dn} = k \,|\, \theta_d'\big) = \sum_{k=1}^{K} \phi_{k w_{dn}} \theta_{dk}' \ . \tag{7.32}$$

Here we note that the distributions ϕ' and θ are integrated out (following the method shown in Appendix A.6).

We calculate the perplexity corresponds to both the training data and test data. Note that the perplexity estimate is unbiased since the words used in estimating θ' are not used for evaluation. We present the perplexity result in Table 7.4, showing the significantly (at 5 % significance level) better performance of CNTM against the baselines on the ML, M10 and AvS datasets. For these datasets, inclusion of citation information also provides additional improvement for model fitting, as shown in the comparison of the CNTM with and without network component. Note that for the CS, Cora and PubMed datasets, the nonparametric ATM was not performed due to the lack of authorship information.

7.7.4 Document Clustering

Next, we evaluate the clustering ability of the topic models. As mentioned in Section 7.6, for M10 and AvS datasets, we assume their ground truth classes correspond to the query categories used in creating the datasets. The ground truth classes for CS, Cora and PubMed datasets were provided. Note we do not use the ML dataset since it has only one category.

We evaluate the clustering performance with *purity* and *normalised mutual information* (NMI), as discussed in Section 5.5.4. Purity is a simple clustering measure which can be interpreted as the proportion of documents correctly clustered, while NMI is an information theoretic measures used for clustering comparison.

Table 7.4: Perplexity for the training and test documents for all datasets, a lower perplexity means better model fitting. We can see that the CNTM with the citation network generally outperforms the other topic models in model fitting. Note that the nonparametric ATM is not performed for the last three datasets due to the lack of authorship information in these datasets.

Model	Perplexity			
	Training	Test	Training	Test
	ML		**M10**	
Bursty HDP-LDA	4904.2 ± 71.3	4992.9 ± 65.6	2467.9 ± 34.8	2825.6 ± 61.4
Nonparametric ATM	2238.2 ± 12.2	2460.3 ± 11.3	1822.4 ± 15.0	2056.4 ± 18.3
CNTM w/o network	2036.3 ± 4.6	2118.1 ± 3.7	922.6 ± 11.0	1263.9 ± 8.8
CNTM w network	**1919.5** ± 8.8	**2039.5** ± 11.7	**910.2** ± 13.3	**1261.0** ± 25.7
	AvS		**CS**	
Bursty HDP-LDA	2460.4 ± 66.4	2612.8 ± 91.7	1498.4 ± 4.1	1616.8 ± 38.8
Nonparametric ATM	2225.9 ± 45.5	2511.9 ± 52.4	N/A	N/A
CNTM w/o network	1540.2 ± 18.5	1959.2 ± 2.4	1506.8 ± 4.4	1609.5 ± 39.2
CNTM w network	**1515.9** ± 2.1	**1938.9** ± 10.4	**1168.6** ± 27.3	**1588.2** ± 93.9
	Cora		**PubMed**	
Bursty HDP-LDA	678.3 ± 1.7	**706.3** ± 16.8	300.0 ± 0.3	300.2 ± 1.2
CNTM w/o network	554.8 ± 14.1	881.1 ± 110.9	**299.9** ± 0.2	300.1 ± 1.3
CNTM w network	**527.0** ± 8.7	719.0 ± 111.4	350.5 ± 20.1	**297.3** ± 3.2

The clustering results are presented in Table 7.5. We can see that the CNTM greatly outperforms the PMTLM in NMI evaluation. Note that for a fair comparison against PMTLM, the experiments on the CS, Cora and PubMed datasets are evaluated with a 10-fold cross validation. Additionally, we would like to point out that since no author information is provided on these 3 datasets, the CNTM becomes a variant of HDP-LDA, but with PYP instead of DP. We find that incorporating supervision into the topic model leads to improvement on clustering task, as predicted. However, this is not the case for the PubMed dataset, we suspect this is because the publications in the PubMed dataset are highly related to one another so the category labels are less useful (see Table 7.3).

7.8 Qualitative Analysis of Learned Topic Models

We move on to perform qualitative analysis on the learned topic models in this section. More specifically, we inspect the learned topic–word distributions, as well as the topics associated with the authors. Additionally, we present a visualisation of the author–topics network learned by the CNTM.

Table 7.5: Comparison of clustering performance on all datasets except the ML dataset, showing the purity and the NMI from the training set, the test set and the overall corpus. Higher purity and NMI indicate better performance. The results presented, in units of 0.01, show that the CNTM and the SCNTM generally outperform the baselines. Note that the best PMTML results are chosen for comparison, which is obtained from Table 2 in Zhu *et al.* [2013]

Dataset	Model		Purity			NMI	
		Train	Test	Overall	Train	Test	Overall
M10	Bursty HDP-LDA	61.7	65.6	62.1	34.8	67.0	38.0
	Nonparametric ATM	55.4	57.8	55.7	29.1	63.0	32.4
	CNTM w/o network	67.3	64.9	67.0	42.5	66.5	44.9
	CNTM w network	66.4	**69.9**	66.8	41.1	**68.6**	43.8
	SCNTM ($\eta = 10$)	85.3	53.1	82.1	60.4	62.7	60.6
	SCNTM ($\eta = \infty$)	**88.1**	47.8	**84.0**	**62.3**	62.3	**62.3**
AvS	Bursty HDP-LDA	72.8	75.0	73.0	32.1	66.3	35.5
	Nonparametric ATM	64.1	65.2	64.2	24.7	61.9	28.4
	CNTM w/o network	77.0	**76.3**	76.9	37.4	66.6	40.3
	CNTM w network	76.0	74.0	75.8	35.4	65.5	38.4
	SCNTM ($\eta = 10$)	**87.9**	67.3	**85.8**	47.5	**66.7**	**49.4**
	SCNTM ($\eta = \infty$)	87.1	50.5	83.4	**47.8**	64.5	**49.4**
CS	PMTLM	N/A	N/A	N/A	N/A	41.4	N/A
	Bursty HDP-LDA	30.5	41.4	31.6	4.9	60.5	10.5
	CNTM w/o network	27.6	41.6	29.0	9.0	61.1	14.2
	CNTM w network	32.6	**44.6**	33.8	13.0	63.4	18.0
	SCNTM ($\eta = \infty$)	**71.2**	33.6	**67.5**	**56.6**	**69.1**	**57.9**
Cora	PMTLM	N/A	N/A	N/A	N/A	51.4	N/A
	Bursty HDP-LDA	31.0	34.4	31.4	3.6	58.5	9.1
	CNTM w/o network	34.0	35.4	34.1	7.7	58.6	12.8
	CNTM w network	37.9	**40.5**	38.2	13.7	61.2	18.5
	SCNTM ($\eta = \infty$)	**86.3**	39.2	**81.6**	**83.5**	**69.8**	**82.2**
PubMed	PMTLM	N/A	N/A	N/A	N/A	27.0	N/A
	Bursty HDP-LDA	49.3	54.3	49.8	9.7	**72.9**	16.0
	CNTM w/o network	53.1	53.7	53.2	15.7	72.5	21.4
	CNTM w network	**54.5**	**54.8**	**54.5**	16.3	72.7	22.0
	SCNTM ($\eta = \infty$)	53.2	53.5	53.2	**16.5**	72.6	**22.2**

7.8.1 Topical Summary of the Datasets

By analysing the topic–word distribution ϕ_k for each topic k, we obtain the topical summary of the datasets. This is achieved by querying the top words associated with each topic k from ϕ_k, which are learned by the CNTM. The top words give us

Table 7.6: Topical summary for the ML, M10 and AvS datasets. The top words are extracted from the topic–word distributions ϕ learned by the CNTM.

Topic	Top Words
	ML
Reinforcement Learning	reinforcement, agents, control, state, task
Object Recognition	face, video, object, motion, tracking
Data Mining	mining, data mining, research, patterns, knowledge
SVM	kernel, support vector, training, clustering, space
Speech Recognition	recognition, speech, speech recognition, audio, hidden markov
	M10
DNA Sequencing	genes, gene, sequence, binding sites, dna
Agriculture	soil, water, content, soils, ground
Financial Market	volatility, market, models, risk, price
Bayesian Modelling	bayesian, methods, models, probabilistic, estimation
Quantum Theory	quantum, theory, quantum mechanics, classical, quantum field
	AvS
Language Modelling	type, polymorphism, types, language, systems
Molecular Structure	copper, protein, model, water, structure
Quantum Theory	theory, quantum, model, quantum mechanics, systems
Social Science	research, development, countries, information, south africa
Family Well-being	children, health, research, social, women

an idea of what the topics are about. In Table 7.6, we display some major topics and the corresponding top words. We note that the topic labels are manually assigned based on the top words. For example, we find that the major topics associated with the ML dataset are various disciplines on machine learning such as reinforcement learning and data mining.

We did not display the topical summary for the CS, Cora and PubMed datasets. The reason being that the original word information is lost in the CS and Cora datasets since the words were converted into integers, which are not meaningful. While for the PubMed dataset, we find that the topics are too similar to each other and thus not interesting. This is mainly because the PubMed dataset is focus on one particular topic, which is on Diabetes Mellitus.

7.8.2 Analysing Authors' Research Area

In CNTM, we model the author–topic distribution ν_i for each author i. This allows us to analyse the topical interest of each author in a collection of publications. Here, we focus on the M10 dataset since it covers a more diverse research areas. For each author i, we can determine their dominant topic k by looking for the largest topic in ν_i. Knowing the dominant topic k of the authors, we can then extract the corresponding top words from the topic–word distribution ϕ_k.

Table 7.7: Major authors and their main research area. Top words are extracted from the topic–word distribution ϕ_k corresponding to the dominant topic k of the author.

Author	Topic	Top Words
D. Aerts	Quantum Theory	quantum, theory, quantum mechanics, classical
Y. Bengio	Neural Network	networks, learning, recurrent, neural
C. Boutilier	Decision Making	decision making, agents, decision, theory, agent
S. Thrun	Robot Learning	robot, robots, control, autonomous, learning
M. Baker	Financial Market	market, risk, firms, returns, financial
E. Segal	Gene Clustering	clustering, processes, gene expression, genes
P. Tabuada	Control System	systems, hybrid, control systems, system, control
L. Ingber	Statistical Mechanic	statistical, mechanics, systems, users, interactions

In Table 7.7, we display the dominant topic associated with several major authors and the corresponding top words. For instance, we can see that the author D. Aerts's main research area is in Quantum theory, while M. Baker focuses on financial market. Again, we note that the topic labels are manually assigned to the authors based on the top words associated with their dominant topics.

7.8.3 Author–topics Network Visualisation

In addition to inspecting the topic and word distributions, we present a way to graphically visualise the author–topics network learned by the CNTM, using Graphviz.[35] On the ML, M10 and AvS datasets, we analyse the influential authors and their connections with the various topics learned by the CNTM. The influential authors are determined based on a measure we call author influence, which is the sum of the λ^- of all their publications, that is, the influence of an author i is

$$\text{Influence}(i) = \sum_{d=1}^{D} \lambda_d^- \, I(a_d = i) \,, \tag{7.33}$$

Note that a_d denotes the author of document d, and $I(\cdot)$ is the indicator function, as previously defined.

Figure 7.2 shows a snapshot of the author–topics network of the ML dataset. The pink rectangles in the snapshot represent the topics learned by the CNTM, showing the top words of the associated topics. The colour intensity (pinkness) of the rectangle shows the relative weight of the topics in the corpus. Connected to the rectangles are ellipses representing the authors, their size is determined by their corresponding author influence in the corpus. For each author, the thickness of the line connecting to a topic shows the relative weight of the topic. Note that not all connections are shown, some of the weak connections are dropped to create a neater

[35] http://www.graphviz.org/ (last accessed 18 August 2014)

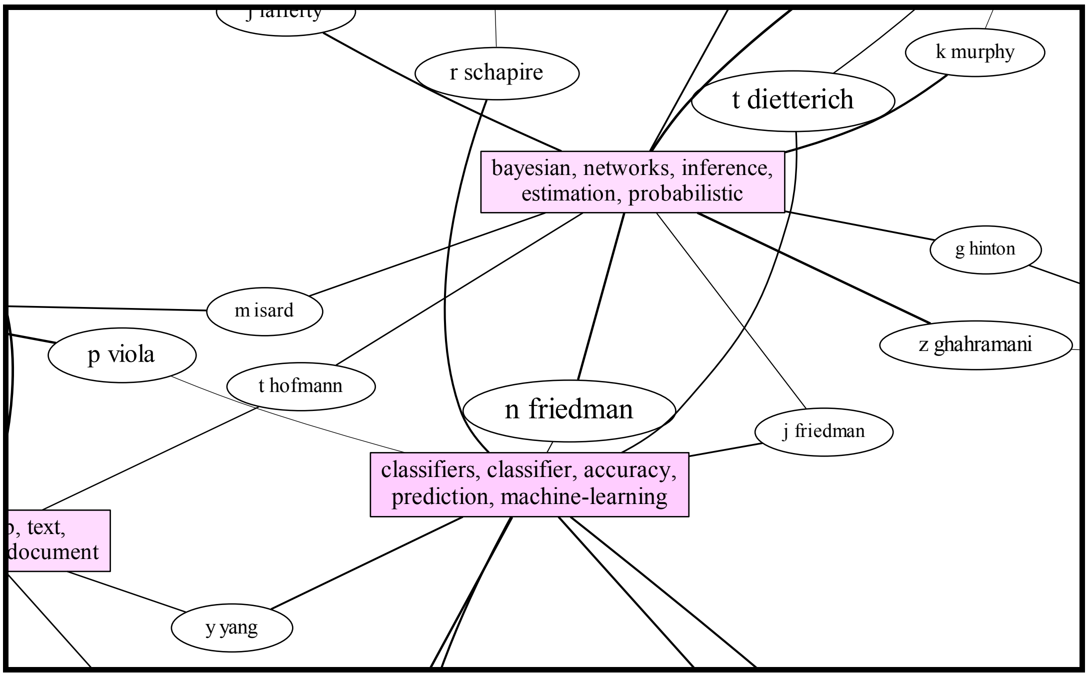


Figure 7.2: Snapshot of the author–topics network from the ML dataset. The pink rectangles represent the learned topics, their intensity (pinkness) corresponds to the topic proportion. The ellipses represent the authors, their size corresponds to the author's influence in the corpus. The strength of the connections are given by the lines' thickness.

diagram. In Figure 7.2, we can see that Z. Ghahramani works mainly in the area of Bayesian inference, as illustrated by the strong connection to the topic with top words "bayesian, networks, inference, estimation, probabilistic". While N. Friedman works in both Bayesian inference and machine learning classification, though with a greater proportion in Bayesian inference. Due to the large size of the plots, we present online[36] the full visualisation of the author–topics network learned from the CiteSeer^X datasets.

7.9 Diagnostics

In this section, we perform some diagnostic tests for the CNTM. We assess the convergence of the MCMC algorithm associated with CNTM and inspect the counts associated with the PYP for the document–topic distributions. Finally, we also present a discussion on the running time of the CNTM.

[36] https://drive.google.com/folderview?id=0B74l2KFRFZJmVXdmbkc3UlpUbzA
(please download and view with a web browser for best quality, last accessed 24 July 2016)

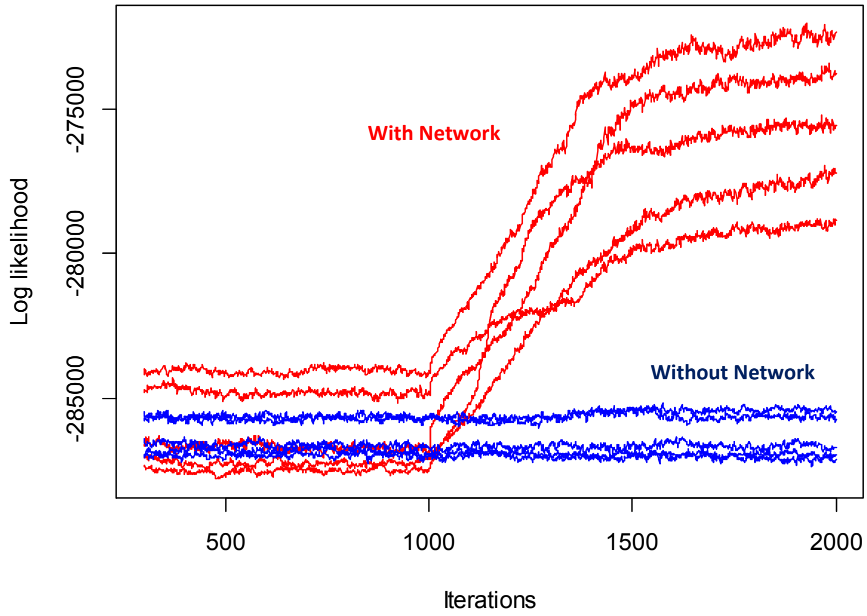


Figure 7.3: Training word log likelihood *vs* iterations during training of the CNTM with and without the network component. The red lines show the log likelihoods of the CNTM with the citation network while the blue lines represent the CNTM without the citation network. The five runs are obtained from five different folds of the Cora dataset.

7.9.1 Convergence Analysis

It is important to assess the convergence of an MCMC algorithm to make sure that the algorithm is not prematurely terminated. In Figure 7.3, we show the time series plot of the training word log likelihood $\sum_{dn} \log p(w_{dn} \,|\, z_{dn}, \phi')$ corresponds to the CNTM trained with and without the network information. Recall that for the CNTM, the Gibbs sampler is first performed for 1,000 iterations before performing the full inference algorithm.

From Figure 7.3, we can clearly see that the Gibbs sampler converges quickly. For the CNTM, it is interesting to see that the log likelihood improves significantly once the network information is used for training (red lines), suggesting that the citation information is useful. Additionally, we like to note that the acceptance rate of the MH algorithm for the citation network averages to about 95 %, which is very high, suggesting that the proposed MH algorithm is effective.

7.9.2 Inspecting Document–topic Hierarchy

As previously mentioned, modelling document–topic hierarchy allows us to balance the contribution of text information and citation information toward topic modelling. In this section, we inspect the customer and table counts associated with the document–topic distributions θ' and θ to give an insight on how the above modelling works. We first note that the number of words in a document tend to be higher than the number of citations.

We illustrate with an example from the ML dataset, we look at the 600th document, which contains 84 words but only 4 citations. The words are assigned to two topics and we have $c_1^\theta = 53$ and $c_2^\theta = 31$. These customer counts are contributed to θ' by way of the corresponding table counts $t_1^\theta = 37$ and $t_2^\theta = 20$. The citations contribute counts directly to θ', in this case, three of the citations are assigned the first topic while another one is assigned to the second topic. The customer count for θ' is the sum of the table counts from θ and the counts from citations. Thus, $c_1^{\theta'} = 37 + 3 = 40$ and $c_2^{\theta'} = 20 + 1 = 21$. Note that the counts from θ' are used to determine the topic composition of the document. By modelling the document–topic hierarchy, we have effectively diluted the influence of text information, this is essential to counter the higher number of words compared to citations.

7.9.3 Computation Complexity

Finally, we briefly discuss the computational complexity of the proposed MCMC algorithm for the CNTM. Although we did not particularly optimise our implementation for speed, the algorithm is of linear time with the number of words, the number of citations and the number of topics. Recall that all implementations are written in the Java programming language.

For the Gibbs sampling algorithm of the hierarchical PYP topic model, as discussed in Chapter 5, we implemented a general Gibbs sampling framework that works with arbitrary PYP network, this allows us to test various PYP topic models with ease and reduce the development time. However, having a general framework for PYP topic models means it is harder to optimise the implementation, thus it performs slower than existing implementations (such as the hca[37]). Nevertheless, the running time is linear with the number of words in the corpus and the number of topics, and constant time with the number of citations.

A naïve implementation of the MH algorithm for the citation network would be of polynomial time, due to the calculation of the double summation in the posterior. However, with caching and reformulation of the double summation, we can evaluate the posterior in linear time. Our implementation of the MH algorithm is linear (in time) with the number of citations and the number of topics, and it is constant time with respect to the number of words.

Table 7.8 shows the average time taken to perform the learning algorithm for 2,000 iterations. All the experiments were performed with a machine having *Intel(R) Core(TM) i7 CPU @ 3.20GHz* (though only 1 processor was used) and *24 Gb RAM*.

[37] http://mloss.org/software/view/527/ (last accessed 18 August 2014)

Table 7.8: Time taken to perform 2,000 iterations of the training algorithm given the statistics of the datasets. The reported SCNTM run time corresponds to $\eta = \infty$. The run time of the SCNTM is of similar magnitude to that of the CNTM.

Datasets	Total Words	Citations	Number of Topics	Time (mins)	
				CNTM	SCNTM
ML	8 270 084	1 105 462	20	16 194	-
M10	595 918	77 222	50	1 772	1 845
AvS	1 102 608	54 601	30	2 131	2 092
CS	105 322	4 608	6	45	43
Cora	49 286	5 429	7	24	26
PubMed	1 332 869	44 335	3	532	397

7.10 Summary

In this chapter, we propose the Citation Network Topic Model (CNTM) to jointly model research publications and their citation network. The CNTM makes use of the author information as well as the categorical labels associated with each document for supervised learning. CNTM performs text modelling with a hierarchical PYP topic model and models the citations with the Poisson distribution given the learned topic distributions. We also propose a novel learning algorithm for the CNTM, which exploits the conjugacy of the Dirichlet distribution and the multinomial distribution, allowing the sampling of the citation networks to be of similar form to the collapsed Gibbs sampler of a topic model. As discussed, our learning algorithm is intuitive and easy to implement.

The CNTM offers substantial performance improvement over previous work [Zhu *et al.*, 2013]. On three CiteSeer^{X} datasets and three existing and publicly available datasets, we demonstrate the improvement of joint topic and network modelling in terms of model fitting and clustering evaluation. Additionally, incorporating supervision into the CNTM provides further improvement on the clustering task. Analysing the learned topic models let us extract useful information on the corpora, for instance, we can inspect the learned topics associated with the documents and examine the research interest of the authors. We also visualise the author–topic network learned by the CNTM, which allows us to have a quick look at the connection between the authors by way of their research areas.

Future work on this area includes learning the influences of the co-authors and utilising them for author merging. In the next chapter, we will go back to modelling tweets, we present a fully Bayesian topic model that jointly models the text content and the underlying social media network between the authors.

Chapter 8

Modelling Text and Author Network on Tweets

This chapter discusses how we make use of auxiliary information that is available on Twitter for a fully Bayesian modelling of tweets. In particular, we incorporate the authors, hashtags, the "follower" network, and the text content. we propose the *Twitter Network Topic model* (TNTM) to jointly model the text and the social network in a fully Bayesian nonparametric way. The TNTM employs a HPYP for text modelling and a GP random function model for social network modelling. We show that the TNTM significantly outperforms several existing nonparametric models due to its flexibility. Moreover, the TNTM enables additional informative inference such as authors' interests, hashtag analysis, as well as leading to further applications such as author recommendation, automatic topic labelling, and hashtag suggestion.

This work differs from Chapter 7 in that we model the network between the authors rather than between the documents. We also emphasise that the treatment on the hashtags is different to that of Chapter 6, here, we model the hashtags directly (treating them as words). This chapter is adapted and extended from Lim *et al.* [2013].

8.1 Introduction

Emergence of web services such as blogs, microblogs and social networking websites allows people to contribute information freely and publicly. This user-generated information is generally more personal, informal, and often contains personal opinions. In aggregate, it can be useful for reputation analysis of entities and products [Aula, 2010], natural disasters detection [Karimi *et al.*, 2013], obtaining first-hand news [Broersma and Graham, 2012], or even demographic analysis [Correa *et al.*, 2010]. In this chapter, we focus on Twitter, an accessible source of information that allows users to freely voice their opinions and thoughts in short text known as tweets.

Although the Latent Dirichlet allocation (LDA) introduced in Section 4.1 is a popular model for text modelling, a direct application on tweets yields poor result as tweets are short and often noisy [Zhao *et al.*, 2011; Baldwin *et al.*, 2013], that is, tweets are unstructured and often contain grammatical and spelling errors, as well as *informal* words such as user-defined abbreviations due to the 140 characters limit. LDA fails on short tweets since it is heavily dependent on word co-occurrence. Also notable is that the text in tweets may contain special tokens known as *hashtags*; they are used as keywords and allow users to link their tweets with other tweets tagged with the same hashtag. Nevertheless, hashtags are informal since they have no standards. Hashtags can be used as both inline words or categorical labels. When used as labels, hashtags are often noisy, since users can create new hashtags easily and use any existing hashtags in any way they like.[38] Hence instead of being hard labels, hashtags are best treated as special words which can be the themes of the tweets. These properties of tweets make them challenging for topic models, and *ad hoc* alternatives are used instead. For instance, Maynard *et al.* [2012] advocate the use of shallow method for tweets, and Mehrotra *et al.* [2013] utilise a tweet-pooling approach to group short tweets into a larger document. In other text analysis applications, tweets are often 'cleansed' by NLP methods such as lexical normalisation [Baldwin *et al.*, 2013]. However, the use of normalisation is also criticised [Eisenstein, 2013], as normalisation can change the meaning of text.

In this chapter, we propose a novel method in modelling microblogs by leveraging the auxiliary information that accompanies tweets. This information, complementing word co-occurrence, allows us to model the tweets better, as well as opening the door to more applications, such as user recommendation and hashtag suggestion. Our major contributions include (1) a fully Bayesian nonparametric model named the Twitter Network Topic model (TNTM) that models tweets well, and (2) a combination of both the HPYP and the GP to jointly model text, hashtags, authors and the followers network. Despite the complexity of the TNTM, its implementation is made relatively straightforward using the flexible framework developed in Chapter 5.

The rest of this chapter is organised as follows. We present some related work in Section 8.2. We then describe the TNTM and its posterior likelihood in Section 8.3 and 8.4. The inference procedures for the TNTM are discussed in Section 8.5. Section 8.6 and 8.7 describe the datasets and the preprocessing procedure. In Section 8.8, we evaluate the TNTM based on several evaluation measures and ablation studies, which show that each part of the TNTM is important. Qualitative results suggest that the learned topics are informative. Finally, we conclude this chapter and outline.

8.2 Related Work

The simplest of Bayesian topic model, LDA, is often extended for different types of data. As discussed, some notable examples are the *author-topic model* (ATM) [Rosen-

[38] For example, *hashtag hijacking*, where a well defined hashtag is used in an "inappropriate" way. The most notable example would be on the hashtag *#McDStories*, though it was initially created to promote happy stories on McDonald's, the hashtag was hijacked with negative stories on McDonald's.

Zvi *et al.*, 2004], the *tag-topic model* [Tsai, 2011], the *supervised LDA* [Mcauliffe and Blei, 2008], and the *Topic-Link LDA* [Liu *et al.*, 2009]. These models only deal with one kind of additional information and thus do not work well with tweets. Note that the tag-topic model treats hashtags as hard labels and uses them to group tweets, which is not appropriate due to the noisy nature of the hashtags.

On the other hand, the *Twitter-LDA* [Zhao *et al.*, 2011] and the *behaviour-topic model* [Qiu *et al.*, 2013] were designed to explicitly model tweets. In contrary to LDA, both models are not admixture models since they impose a limit of only one topic per document. The behaviour-topic model analyses tweets' posting behaviour[39] for each topic, and uses them for user recommendation. Alternatively, the *biterm topic model* [Yan *et al.*, 2013] uses only the biterm co-occurrences to model tweets, discarding document level information. Both the biterm topic model and the Twitter-LDA do not incorporate any auxiliary information. All the mentioned topic models also have a limitation in that the number of topics need to be specified in advance, which is difficult since this number is not known.

Some recent work makes use of the link between documents (*e.g.*, citations) in topic modelling, including the CNTM in Chapter 7, the *relational topic model* [Chang and Blei, 2010], the *Poisson mixed-topic link model* [Zhu *et al.*, 2013] and the *Link-PLSA-LDA* [Nallapati *et al.*, 2008]. Some other work models the authors' network information, such as the *Topic-Link LDA*, which models author community using a generalised linear model, and the *Author Cite Topic Model* [Kataria *et al.*, 2011], which models the authors citation network. However, these models are parametric in nature and can be restrictive. On the contrary, Lloyd *et al.* [2012] use a very flexible nonparametric model for network data by utilising random function priors, but they do not model text. We note that the TNTM makes use of the random function network model of Lloyd *et al.* [2012], but we apply modifications to the random function network model that leads to significant model improvement, this is discussed in the next section.

8.3 The Twitter Network Topic model

The TNTM makes use of the accompanying *hashtags*, *authors*, and *followers network* to model tweets better. The TNTM is composed of two main components: a HPYP topic model for the text and hashtags, and a GP based random function network model for the followers network. The authorship information serves to connect the two together. The HPYP topic model is illustrated by region ⓑ in Figure 8.1 while the network model is captured by region ⓐ.

[39] Whether they are original post or retweet.

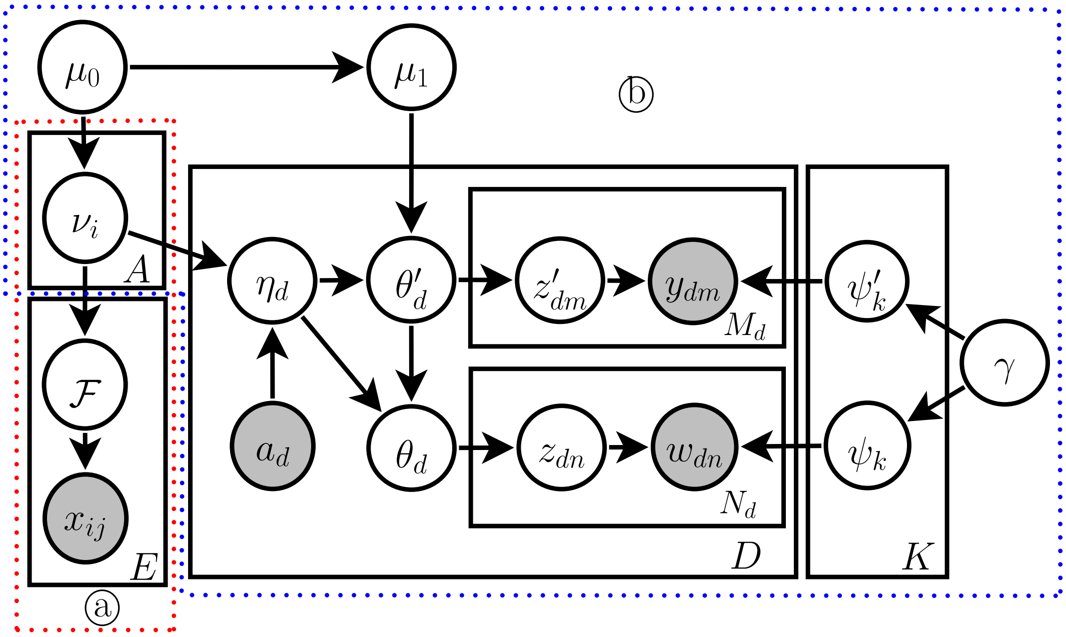


Figure 8.1: Graphical model for the Twitter Network Topic model (TNTM). The latent variables are unshaded and the observed variables are shaded. The TNTM is composed of a HPYP topic model (region ⓑ) and a GP based random function network model (region ⓐ). The author–topic distributions ν serve to link the two together. Each tweet is modelled with a hierarchy of document–topic distributions denoted by η, θ', and θ, where each is attuned to the whole tweet, the hashtags, and the words, in that order. With their own topic assignments z' and z, the hashtags y and the words w are separately modelled. They are generated from the topic–hashtag distributions ψ' and the topic–word distributions ψ respectively. The variables μ_0, μ_1 and γ are priors for the respective PYPs. The connections between the authors are denoted by x, which are modelled by random function $\mathcal{F}$.

8.3.1 HPYP Topic Model

We design the HPYP topic model as follows. For the word distributions, we first generate a parent word distribution prior γ for all topics:

$$\gamma \sim \mathrm{PYP}(\alpha^{\gamma}, \beta^{\gamma}, H^{\gamma}) \,, \tag{8.1}$$

where H_γ is a discrete uniform distribution over the complete word vocabulary $\mathcal{V}$.[40] Then, we sample the hashtag distribution ψ'_k and word distribution ψ_k for each

[40] The complete word vocabulary contains words and hashtags seen in the corpus.

topic k, with γ as the base distribution:

$$\psi'_k \,|\, \gamma \sim \mathrm{PYP}(\alpha^{\psi'_k}, \beta^{\psi'_k}, \gamma)\,, \tag{8.2}$$

$$\psi_k \,|\, \gamma \sim \mathrm{PYP}(\alpha^{\psi_k}, \beta^{\psi_k}, \gamma)\,, \qquad \text{for } k = 1, \dots, K\,. \tag{8.3}$$

Note that the tokens of the hashtags are shared with the words, that is, the hashtag *#happy* shares the same token as the word *happy*, and are thus treated as the same word. This treatment is important since some hashtags are used as words instead of labels.[41] Additionally, this also allows any words to be hashtags, which will be useful for hashtag recommendation.

For the topic distributions, we generate a global topic distribution μ_0 that serves as a prior. Then generate the author–topic distribution ν_i for each author i, and a miscellaneous topic distribution μ_1 to capture topics that deviate from the authors' usual topics:

$$\mu_0 \sim \mathrm{GEM}(\alpha^{\mu_0}, \beta^{\mu_0})\,, \tag{8.4}$$

$$\mu_1 \,|\, \mu_0 \sim \mathrm{PYP}(\alpha^{\mu_1}, \beta^{\mu_1}, \mu_0)\,, \tag{8.5}$$

$$\nu_i \,|\, \mu_0 \sim \mathrm{PYP}(\alpha^{\nu_i}, \beta^{\nu_i}, \mu_0)\,, \qquad \text{for } i = 1, \dots, A\,. \tag{8.6}$$

For each tweet d, given ν and the observed author a_d, we sample the document–topic distribution η_d, as follows:

$$\eta_d \,|\, a_d, \nu \sim \mathrm{PYP}(\alpha^{\eta_d}, \beta^{\eta_d}, \nu_{a_d})\,, \qquad \text{for } d = 1, \dots, D\,. \tag{8.7}$$

Next, we generate the topic distributions for the observed hashtags (θ'_d) and the observed words (θ_d), following the technique used in the adaptive topic model [Du *et al.*, 2012a]. We explicitly model the influence of hashtags to words, by generating the words conditioned on the hashtags. The intuition comes from hashtags being the themes of a tweet, and they drive the content of the tweet. Specifically, we sample the mixing proportions $\rho^{\theta'_d}$, which control the contribution of η_d and μ_1 for the base distribution of θ'_d, and then generate θ'_d given $\rho^{\theta'_d}$:

$$\rho^{\theta'_d} \sim \mathrm{Beta}\Big(\lambda_0^{\theta'_d}, \lambda_1^{\theta'_d}\Big)\,, \tag{8.8}$$

$$\theta'_d \,|\, \mu_1, \eta_d \sim \mathrm{PYP}\Big(\alpha^{\theta'_d}, \beta^{\theta'_d}, \rho^{\theta'_d}\mu_1 + (1-\rho^{\theta'_d})\eta_d\Big)\,. \tag{8.9}$$

We set θ'_d and η_d as the parent distributions of θ_d. This flexible configuration allows us to investigate the relationship between θ_d, θ'_d and η_d, that is, we can examine if θ_d is directly determined by η_d, or through the θ'_d. The mixing proportions ρ^{θ_d} and the

[41] For instance, as illustrated by the following tweet: *i want to get into #photography. can someone recommend a good beginner #camera please? i dont know where to start?*

topic distribution θ_d is generated similarly:

$$\rho^{\theta_d} \sim \mathrm{Beta}\Big(\lambda_0^{\theta_d}, \lambda_1^{\theta_d}\Big) \,, \tag{8.10}$$

$$\theta_d \,|\, \eta_d, \theta'_d \sim \mathrm{PYP}\Big(\alpha^{\theta_d}, \beta^{\theta_d}, \, \rho^{\theta_d}\eta_m + (1-\rho^{\theta_d})\theta'_d\Big) \,. \tag{8.11}$$

The hashtags and words are then generated in a similar fashion to LDA. For the m-th hashtag in tweet d, we sample a topic z'_{dm} and the hashtag y_{dm} by

$$z'_{dm} \,|\, \theta'_d \sim \mathrm{Discrete}\big(\theta'_d\big) \,, \tag{8.12}$$

$$y_{dm} \,|\, z'_{dm}, \psi' \sim \mathrm{Discrete}\Big(\psi'_{z'_{dm}}\Big) \,, \qquad \text{for } \; m = 1, \ldots, M_d \,, \tag{8.13}$$

where M_d is the number of seen hashtags in tweet d. While for the n-th word in tweet d, we sample a topic z_{dn} and the word w_{dn} by

$$z_{dn} \,|\, \theta_d \sim \mathrm{Discrete}(\theta_d) \,, \tag{8.14}$$

$$w_{dn} \,|\, z_{dn}, \psi \sim \mathrm{Discrete}\big(\psi_{z_{dn}}\big) \,, \qquad \text{for } \; n = 1, \ldots, N_d \,, \tag{8.15}$$

where N_d is the number of observed words in tweet d. We note that all above α, β and λ are the hyperparameters of the model. We show the importance of the above modelling with ablation studies in Section 8.8. Although the HPYP topic model may seem complex, it is actually a simple network of PYP nodes since all distributions on the probability vectors are modelled by the PYP. The advantage of such modelling was discussed in Chapter 5.

8.3.2 Random Function Network Model

The network modelling is connected to the HPYP topic model *via* the author–topic distributions ν, where we treat ν as inputs to the GP in the network model. The GP, represented by $\mathcal{F}$, determines the link between two authors (x_{ij}), which indicates the existence of the social links between author i and author j. For each pair of authors, we sample their connections with the following random function network model:

$$Q_{ij} \,|\, \nu \sim \mathcal{F}(\nu_i, \nu_j) \,, \tag{8.16}$$

$$x_{ij} \,|\, Q_{ij} \sim \mathrm{Bernoulli}\big(s(Q_{ij})\big) \,, \qquad \text{for } \; i = 1, \ldots, A; \; j = 1, \ldots, A \,, \tag{8.17}$$

where $s(\cdot)$ is the *sigmoid function*:

$$s(t) = \frac{1}{1 + e^{-t}} \,. \tag{8.18}$$

By marginalising out $\mathcal{F}$, we can write $\mathbf{Q} \sim \mathrm{GP}(\varsigma, \kappa)$, where $\mathbf{Q}$ is a *vectorised* collection of Q_{ij},[42] ς denotes the mean vector and κ is the covariance matrix of the GP:

$$\varsigma_{ij} = \mathrm{Sim}(\nu_i, \nu_j)\,, \tag{8.19}$$

$$\kappa_{ij,i'j'} = \frac{s^2}{2}\exp\left(-\frac{\left|\mathrm{Sim}(\nu_i,\nu_j) - \mathrm{Sim}(\nu_{i'},\nu_{j'})\right|^2}{2l^2}\right) + \sigma^2 I(ij = i'j')\,, \tag{8.20}$$

where s, l and σ are the hyperparameters associated to the kernel. $\mathrm{Sim}(\cdot,\cdot)$ is a similarity function that has a range between 0 and 1. In this chapter, we choose the *cosine similarity* the similarity function due to ease of computation and its popularity in natural language processing:

$$\mathrm{Sim}(\nu_i, \nu_j) = \frac{\nu_i \cdot \nu_j}{|\nu_i||\nu_j|}\,. \tag{8.21}$$

Note that our kernel definition is different from Lloyd *et al.* [2012]. We define the kernel function such that the authors with similar topics are connected, while the original definition fails to consider the relation between author–topic distributions. We present the list of variables used by the TNTM in Table 8.1.

8.3.3 Relationships with Other Models

The TNTM is related to many existing models after removing certain components of the model. When hashtags and the network components are removed, the TNTM is reduced to a nonparametric variant of the ATM. Oppositely, if authorship information is discarded, the TNTM resembles the *correspondence LDA* [Blei and Jordan, 2003], although it differs in that it allows hashtags and words to be generated from a common vocabulary.

In contrast to existing parametric models, the network model in the TNTM provides possibly the most flexible way of network modelling *via* a nonparametric Bayesian prior (GP), following Lloyd *et al.* [2012]. Different to Lloyd *et al.* [2012], we propose a new kernel function that fits our purpose better and achieves significant improvement over the original kernel function. Moreover, we jointly model the GP and the HPYP, which brings significant challenges for posterior inference.

8.4 Representation and Model Likelihood

As with previous chapters, we represent the TNTM using the CRP representation discussed in Section 5.3. However, since the PYP variables in the TNTM can have multiple parents, we extend the representation following Du *et al.* [2012a]. The distinction is that we store multiple tables counts for each PYP, to illustrate, $t_k^{\mathcal{N}\to\mathcal{P}}$ represents

[42] $\mathbf{Q} = (Q_{11}, Q_{12}, \ldots, Q_{AA})^{\mathsf{T}}$, note that ς and κ follow the same indexing.

Table 8.1: List of variables for the Twitter Network Topic model (TNTM).

Variable	Name	Description
z_{dn}	Word topic	Topic label for word w_{dn}.
z'_{dm}	Hashtag topic	Topic label for hashtag y_{dm}.
w_{dn}	Word	The n-th observed word in document d.
y_{dm}	Hashtag	The m-th observed hashtag in document d.
x_{ij}	Link	Binary variable on author i following author j.
Q_{ij}	Link strength	Strength for the link x_{ij}.
a_d	Author	Author for document d.
ψ_k	Topic–word distribution	Probability distribution in generating words for topic k.
ψ'_k	Topic–hashtag distribution	Probability distribution in generating hashtags for topic k.
θ_d	Document–topic distribution	Probability distribution in generating word topics for document d.
θ'_d	Document–topic distribution	Probability distribution in generating hashtag topics for document d.
η_d	Document–topic prior	Topic prior for θ'_d and θ_d.
ν_a	Author–topic distribution	Probability distribution in generating topics for author a.
γ	Word/hashtag distribution	Word or hashtag prior for ψ_k and ψ'_k.
μ_1	Miscellaneous topic distribution	Topic prior for θ'_d.
μ_0	Global topic distribution	Topic prior for ν_a and μ_1.
$\alpha^{\mathcal{N}}$	Discount	Discount parameter of the PYP $\mathcal{N}$.
$\beta^{\mathcal{N}}$	Concentration	Concentration parameter of the PYP $\mathcal{N}$.
$H^{\mathcal{N}}$	Base distribution	Base distribution of the PYP $\mathcal{N}$.
$\rho^{\mathcal{N}}$	Mixing proportion	Mixing proportion for the base distribution of PYP $\mathcal{N}$.
$\lambda^{\mathcal{N}}$	Shape	Shape parameter for $\rho^{\mathcal{N}}$.
ς	Mean function	Mean function for generating $\mathbf{Q}$.
κ	Covariance	Covariance function for generating $\mathbf{Q}$.

the number of tables in PYP $\mathcal{N}$ serving dish k that are contributed to the customer counts in PYP $\mathcal{P}$, $c_k^{\mathcal{P}}$. Similarly, the total table counts that contribute to $\mathcal{P}$ is denoted as $T^{\mathcal{N}\to\mathcal{P}} = \sum_k t_k^{\mathcal{N}\to\mathcal{P}}$. Note the number of tables in PYP $\mathcal{N}$ is $t_k^{\mathcal{N}} = \sum_{\mathcal{P}} t_k^{\mathcal{N}\to\mathcal{P}}$, while the total number of tables is $T^{\mathcal{N}} = \sum_{\mathcal{P}} T^{\mathcal{N}\to\mathcal{P}}$.

Again, we use bold face capital letters to denote the set of all relevant lower case variables. For example, we denote **W** and **Y** as the set of all words and hashtags; **Z** and **Z**′ as the set of all topic assignments for the words and the hashtags; **T** as the set of all table counts and **C** as the set of all customer counts; and we introduce Ξ as the set of all hyperparameters. By marginalising out the latent variables, we write down the model likelihood corresponding to the HPYP topic model in terms of the counts:

$$\begin{aligned} p(\mathbf{Z},\mathbf{Z}',\mathbf{T},\mathbf{C}\,|\,\mathbf{W},\mathbf{Y},\Xi) &\propto p(\mathbf{Z},\mathbf{Z}',\mathbf{W},\mathbf{Y},\mathbf{T},\mathbf{C}\,|\,\Xi) \\ &\propto f(\mu_0)f(\mu_1)\left(\prod_{i=1}^{A} f(\nu_i)\right)\left(\prod_{k=1}^{K} f(\psi_k')f(\psi_k)\right)f(\gamma) \\ &\quad\times\left(\prod_{d=1}^{D} f(\eta_d)f(\theta_d')f(\theta_d)g(\rho^{\theta_d'})g(\rho_d^{\theta})\right)\left(\prod_{v=1}^{|\mathcal{V}|}\left(\frac{1}{|\mathcal{V}|}\right)^{t_v^{\gamma}}\right), \end{aligned} \tag{8.22}$$

where $f(\mathcal{N})$ is the modularised likelihood corresponding to node $\mathcal{N}$, as defined by Equation (5.9), and $g(\rho)$ is the likelihood corresponding to the probability ρ that controls which parent node to send a customer to. These likelihoods are defined as

$$f(\mathcal{N}) = \frac{(\beta^{\mathcal{N}}|\alpha^{\mathcal{N}})_{T^{\mathcal{N}}}}{(\beta^{\mathcal{N}})_{C^{\mathcal{N}}}}\prod_{k=1}^{K} S^{c_k^{\mathcal{N}}}_{t_k^{\mathcal{N}},\alpha^{\mathcal{N}}}\binom{c_k^{\mathcal{N}}}{t_k^{\mathcal{N}}}^{-1}, \tag{8.23}$$

$$g(\rho^{\mathcal{N}}) = B\Big(\lambda_0^{\mathcal{N}} + T^{\mathcal{N}\to\mathcal{P}_0},\ \lambda_1^{\mathcal{N}} + T^{\mathcal{N}\to\mathcal{P}_1}\Big), \tag{8.24}$$

for $\mathcal{N} \sim \mathrm{PYP}\big(\alpha^{\mathcal{N}},\beta^{\mathcal{N}},\rho^{\mathcal{N}}\mathcal{P}_0 + (1-\rho^{\mathcal{N}})\mathcal{P}_1\big)$. Note that $(x)_T$ and $(x|y)_T$ denote the Pochhammer symbol, and $S^x_{y,a}$ is the generalised Stirling number, as discussed in Section 5.3. Recall that $B(a,b)$ denotes the beta function that normalises a Dirichlet distribution, defined as follows:

$$B(a,b) = \frac{\Gamma(a)\,\Gamma(b)}{\Gamma(a+b)}\,. \tag{8.25}$$

Note that in Equation (8.22), the topic assignments **Z** are implicitly captured by the following customer counts:

$$c_k^{\theta_d} = \sum_{n=1}^{N_d} I(z_{dn} = k)\,, \tag{8.26}$$

while the topic assignments $\mathbf{Z}'$ are implicitly captured by

$$c_k^{\theta'_d} = t_k^{\theta_d \rightarrow \theta'_d} + \sum_{m=1}^{M_d} I(z_{dm} = k) \,. \tag{8.27}$$

For the random function network model, the conditional posterior can be derived as

$$\begin{aligned} p(\mathbf{Q} \,|\, \mathbf{X}, \nu, \Xi) &\propto p(\mathbf{X}, \mathbf{Q} \,|\, \nu, \Xi) \\ &\propto \Bigg(\prod_{i=1}^{A} \prod_{j=1}^{A} p(x_{ij} \,|\, Q_{ij}) \Bigg) p(\mathbf{Q} \,|\, \nu, \Xi) \\ &\propto \Bigg(\prod_{i=1}^{A} \prod_{j=1}^{A} s(Q_{ij})^{x_{ij}} \Big(1 - s(Q_{ij})\Big)^{1-x_{ij}} \Bigg) \\ &\qquad \times |\kappa|^{-\frac{1}{2}} \exp\bigg(-\frac{1}{2} (\mathbf{Q} - \varsigma)^{\mathsf{T}} \kappa^{-1} (\mathbf{Q} - \varsigma) \bigg) \,. \end{aligned} \tag{8.28}$$

The full posterior likelihood is thus the product of the topic model posterior likelihood (Equation (8.22)) and the network posterior likelihood (Equation (8.28)):

$$p(\mathbf{Q}, \mathbf{Z}, \mathbf{Z}', \mathbf{T}, \mathbf{C} \,|\, \mathbf{X}, \mathbf{W}, \mathbf{Y}, \Xi) = p(\mathbf{Z}, \mathbf{Z}', \mathbf{T}, \mathbf{C} \,|\, \mathbf{W}, \mathbf{Y}, \Xi) \, p(\mathbf{Q} \,|\, \mathbf{X}, \nu, \Xi) \,. \tag{8.29}$$

8.5 Performing Posterior Inference on the TNTM

In the TNTM, combining a GP with a HPYP makes its posterior inference non-trivial. Hence, we employ approximate inference by alternatively performing MCMC sampling on the HPYP topic model and the network model, conditioned on each other. For the HPYP topic model, we employ the flexible framework discussed in Chapter 5 to perform collapsed blocked Gibbs sampling. For the network model, we derive a Metropolis-Hastings (MH) algorithm based on the elliptical slice sampler [Murray *et al.*, 2010]. In addition, the author–topic distributions ν connecting the HPYP and the GP are sampled with an MH scheme since their posteriors do not follow a standard form. We note that the PYPs in this chapter can have multiple parents, so we extend the framework in Chapter 5 to allow for this.

8.5.1 Collapsed Blocked Gibbs Sampler for the HPYP Topic Model

The collapsed Gibbs sampling for the HPYP topic model in TNTM is similar to the procedure in Section 5.4, although there are two main differences. The first difference is that we need to sample the topics for both words and hashtags, each with a different conditional posterior compared to that of Section 5.4. While the second is due to the PYPs in TNTM can have multiple parents, thus an alternative to decrementing the counts is required. Below, we discuss the differences in the inference procedure.

8.5.1.1 Decrementing the Counts Associated with a Word or a Hashtag

When we remove a word or a hashtag during inference, we decrement by one the customer count from the PYP associated with the word or the hashtag, that is, $c_k^{\theta_d}$ for word w_{dn} ($z_{dn} = k$) and $c_k^{\theta'_d}$ for hashtag y_{dm} ($z'_{dm} = k$). Decrementing the customer count may or may not decrement the respective table count. However, if the table count is decremented, then we would decrement the customer count of the parent PYP. This is relatively straightforward in Section 5.4.1 since the PYPs have only one parent. Here, when a PYP $\mathcal{N}$ has multiple parents, we would sample for one of its parent PYPs and decrement the table count corresponding to the parent PYP. Although not the same, the rationale of this procedure follows Section 5.4.1.

We explain in more details below. When the customer count $c_k^{\mathcal{N}}$ is decremented, we introduce an auxiliary variable $u_k^{\mathcal{N}}$ that indicates which parent of $\mathcal{N}$ to remove a table from, or none at all. The sample space for $u_k^{\mathcal{N}}$ is the P parent nodes $\mathcal{P}_1, \ldots, \mathcal{P}_P$ of $\mathcal{N}$, plus $\emptyset$. When $u_k^{\mathcal{N}}$ is equal to $\mathcal{P}_i$, we decrement the table count $t_k^{\mathcal{N}\to\mathcal{P}_i}$ and subsequently decrement the customer count $c_k^{\mathcal{P}_i}$ in node $\mathcal{P}_i$. If $u_k^{\mathcal{N}}$ equals to $\emptyset$, we do not decrement any table count. The process is repeated recursively as long as a customer count is decremented, that is, we stop when $u_k^{\mathcal{N}} = \emptyset$.

The value of $u_k^{\mathcal{N}}$ is sampled as follows:

$$p(u_k^{\mathcal{N}}) = \begin{cases} t_k^{\mathcal{N}\to\mathcal{P}_i} / c_k^{\mathcal{N}} & \text{if } u_k^{\mathcal{N}} = \mathcal{P}_i \\ 1 - \sum_{\mathcal{P}_i} p(u_k^{\mathcal{N}} = \mathcal{P}_i) & \text{if } u_k^{\mathcal{N}} = \emptyset \ . \end{cases} \tag{8.30}$$

To illustrate, when a word w_{dn} (with topic z_{dn}) is removed, we decrement $c_{z_{dn}}^{\theta_d}$, that is, $c_{z_{dn}}^{\theta_d}$ becomes $c_{z_{dn}}^{\theta_d} - 1$. We then determine if this word contributes to any table in node θ_d by sampling $u_{z_{dn}}^{\theta_d}$ from Equation (8.30). If $u_{z_{dn}}^{\theta_d} = \emptyset$, we do not decrement any table count and proceed with the next step in Gibbs sampling; otherwise, $u_{z_{dn}}^{\theta_d}$ can either be θ'_d or η_d, in these cases, we would decrement $t_{z_{dn}}^{\theta_d \to u_{z_{dn}}^{\theta_d}}$ and $c_{z_{dn}}^{u_{z_{dn}}^{\theta_d}}$, and continue the process recursively.

We present the decrementing process in Algorithm 8.1. To remove a word w_{dn} during inference, we would need to decrement the counts contributed by w_{dn} (and z_{dn}). For the topic side, we decrement the counts associated with node $\mathcal{N} = \theta_d$ with group $k = z_{dn}$ using Algorithm 8.1. While for the vocabulary side, we decrement the counts associated with the node $\mathcal{N} = \psi_{z_{dn}}$ with group $k = w_{dn}$. The effect of the word on the other PYP variables are implicitly considered through recursion.

Note that the procedure to decrementing a hashtag y_{dm} is similar, in this case, we decrement the counts for $\mathcal{N} = \theta'_d$ with $k = z'_{dm}$ (topic side), then decrement the counts for $\mathcal{N} = \psi'_{z'_{dm}}$ with $k = y_{dm}$ (vocabulary side).

Algorithm 8.1 Decrementing counts associated with a PYP node $\mathcal{N}$ and group k.

```
1. Decrement the customer count $c_k^{\mathcal{N}}$ by one.
2. Sample an auxiliary variable $u_k^{\mathcal{N}}$ with Equation (8.30).
3. For the sampled $u_k^{\mathcal{N}}$, perform the following:
    (a) If $u_k^{\mathcal{N}} = \emptyset$, exit the algorithm.
    (b) Otherwise, decrement the table count $t_k^{\mathcal{N} \to u_k^{\mathcal{N}}}$ by one and repeat Steps 2 – 4
        by replacing $\mathcal{N}$ with $u_k^{\mathcal{N}}$.
```

8.5.1.2 Sampling a New Topic for a Word or a Hashtag

After decrementing, we sample a new topic for the word or the hashtag. The sampling process follows the procedure discussed in Section 5.4.2, but with different conditional posteriors (for both the word and the hashtag). The full conditional posterior probability for the collapsed blocked Gibbs sampling can be derived easily. For instance, the conditional posterior for sampling the topic z_{dn} of word w_{dn} is

$$p(z_{dn}, \mathbf{T}, \mathbf{C} \,|\, \mathbf{Z}^{-dn}, \mathbf{Z}', \mathbf{W}, \mathbf{Y}, \mathbf{T}^{-dn}, \mathbf{C}^{-dn}, \Xi) = \frac{p(\mathbf{Z}, \mathbf{Z}', \mathbf{T}, \mathbf{C} \,|\, \mathbf{W}, \mathbf{Y}, \Xi)}{p(\mathbf{Z}^{-dn}, \mathbf{Z}', \mathbf{T}^{-dn}, \mathbf{C}^{-dn} \,|\, \mathbf{W}, \mathbf{Y}, \Xi)}\,, \tag{8.31}$$

which can then be easily decomposed into simpler form (see discussion in Section 5.4.2) using Equation (8.22). Here, the superscript $\square^{-dn}$ indicates the word w_{dn} and the topic z_{dn} are removed from the respective sets. Similarly, the conditional posterior probability for sampling the topic z'_{dm} of hashtag y_{dm} can be derived as

$$p(z'_{dm}, \mathbf{T}, \mathbf{C} \,|\, \mathbf{Z}, \mathbf{Z}'^{-dm}, \mathbf{W}, \mathbf{Y}, \mathbf{T}^{-dm}, \mathbf{C}^{-dm}, \Xi) = \frac{p(\mathbf{Z}, \mathbf{Z}', \mathbf{T}, \mathbf{C} \,|\, \mathbf{W}, \mathbf{Y}, \Xi)}{p(\mathbf{Z}, \mathbf{Z}'^{-dm}, \mathbf{T}^{-dm}, \mathbf{C}^{-dm} \,|\, \mathbf{W}, \mathbf{Y}, \Xi)}\,, \tag{8.32}$$

where the superscript $\square^{-dm}$ signals the removal of the hashtag y_{dm} and the topic z'_{dm}.

As in Section 5.4.2, we compute the posterior for all possible changes to **T** and **C** corresponding to the new topic (for z_{dn} or z'_{dm}). We then sample the next state using a Gibbs sampler.

8.5.2 Estimating the Probability Vectors of PYPs with Multiple Parents

Following Section 5.4.4, we estimate the various probability distributions of the PYPs by their posterior means. For a PYP $\mathcal{N}$ with a single PYP parent $\mathcal{P}_1$, as discussed in

Section 5.4.4, we can estimate its probability vector $\hat{\mathcal{N}} = (\hat{\mathcal{N}}_1, \ldots, \hat{\mathcal{N}}_K)$ as

$$\begin{aligned}\hat{\mathcal{N}}_k &= \mathbb{E}[\mathcal{N}_k \,|\, \mathbf{Z}, \mathbf{Z}', \mathbf{W}, \mathbf{Y}, \mathbf{T}, \mathbf{C}, \Xi] \\ &= \frac{\big(\alpha^{\mathcal{N}} T^{\mathcal{N}} + \beta^{\mathcal{N}}\big)\mathbb{E}[\mathcal{P}_{1k} \,|\, \mathbf{Z}, \mathbf{Z}', \mathbf{W}, \mathbf{Y}, \mathbf{T}, \mathbf{C}, \Xi] + c_k^{\mathcal{N}} - \alpha^{\mathcal{N}} T_k^{\mathcal{N}}}{\beta^{\mathcal{N}} + C^{\mathcal{N}}}\,,\end{aligned} \tag{8.33}$$

which lets one analyse the probability vectors in a topic model using recursion.

Unlike the above, the posterior mean is slightly more complicated for a PYP $\mathcal{N}$ that has multiple PYP parents $\mathcal{P}_1, \ldots, \mathcal{P}_P$. Formally, we define the PYP $\mathcal{N}$ as

$$\mathcal{N} \,|\, \mathcal{P}_1, \ldots, \mathcal{P}_P \sim \mathrm{PYP}\Big(\alpha^{\mathcal{N}}, \beta^{\mathcal{N}}, \rho_1^{\mathcal{N}} \mathcal{P}_1 + \cdots + \rho_P^{\mathcal{N}} \mathcal{P}_P\Big)\,, \tag{8.34}$$

where the mixing proportion $\rho^{\mathcal{N}} = (\rho_1^{\mathcal{N}}, \ldots, \rho_P^{\mathcal{N}})$ follows a Dirichlet distribution with parameter $\lambda^{\mathcal{N}} = (\lambda_1^{\mathcal{N}}, \ldots, \lambda_P^{\mathcal{N}})$:

$$\rho^{\mathcal{N}} \sim \mathrm{Dirichlet}\big(\lambda^{\mathcal{N}}\big)\,. \tag{8.35}$$

Before we can estimate the probability vector, we first estimate the mixing proportion with its posterior mean given the customer counts and table counts:

$$\hat{\rho}_i^{\mathcal{N}} = \mathbb{E}[\rho_i^{\mathcal{N}} \,|\, \mathbf{Z}, \mathbf{Z}', \mathbf{W}, \mathbf{Y}, \mathbf{T}, \mathbf{C}, \Xi] = \frac{T^{\mathcal{N} \to \mathcal{P}_i} + \lambda_i^{\mathcal{N}}}{T^{\mathcal{N}} + \sum_i \lambda_i^{\mathcal{N}}}\,. \tag{8.36}$$

Then, we can estimate the probability vector $\hat{\mathcal{N}} = (\hat{\mathcal{N}}_1, \ldots, \hat{\mathcal{N}}_K)$ by

$$\hat{\mathcal{N}}_k = \frac{\big(\alpha^{\mathcal{N}} T^{\mathcal{N}} + \beta^{\mathcal{N}}\big)\hat{H}_k^{\mathcal{N}} + c_k^{\mathcal{N}} - \alpha^{\mathcal{N}} T_k^{\mathcal{N}}}{\beta^{\mathcal{N}} + C^{\mathcal{N}}}\,, \tag{8.37}$$

where $\hat{H}^{\mathcal{N}} = (\hat{H}_1^{\mathcal{N}}, \ldots, \hat{H}_K^{\mathcal{N}})$ is the expected base distribution:

$$\hat{H}_k^{\mathcal{N}} = \sum_{i=1}^{P} \hat{\rho}_i^{\mathcal{N}} \mathbb{E}[\mathcal{P}_{ik} \,|\, \mathbf{Z}, \mathbf{Z}', \mathbf{W}, \mathbf{Y}, \mathbf{T}, \mathbf{C}, \Xi]\,. \tag{8.38}$$

We note that a PYP with a single parent is simply a special case, which can be achieved by setting $\rho_1^{\mathcal{N}}$ to one and the other $\rho_i^{\mathcal{N}}$ to zero.

With these formulations, all the topic distributions and the word distributions in the TNTM can be reconstructed from the customer counts and table counts. For instance, the author–topic distribution ν_i of each author i can be determined recursively by first estimating the topic distribution μ_0. The word distributions for each topic are similarly estimated.

8.5.3 MH Algorithm for the Random Function Network Model

Here, we discuss how we learn the topic distributions μ_0 and ν from the random function network model. We configure the MH algorithm to start after running one thousand iterations of the collapsed blocked Gibbs sampler, this is to we can quickly initialise the TNTM with the HPYP topic model before running the full algorithm. In addition, this allows us to demonstrate the improvement to the TNTM due to the random function network model.

To facilitate the MH algorithm, we have to represent the topic distributions μ_0 and ν explicitly as probability vectors, that is, we do not store the customer counts and table counts for μ_0 and ν after starting the MH algorithm. In the MH algorithm, we propose new samples for μ_0 and ν, and then accept or reject the samples. The details for the MH algorithm is as follow.

In each iteration of the MH algorithm, we use the Dirichlet distributions as proposal distributions for μ_0 and ν:

$$q(\mu_0^{\text{new}} \,|\, \mu_0) = \text{Dirichlet}(\beta^{\mu_0}\mu_0)\,, \tag{8.39}$$

$$q(\nu_i^{\text{new}} \,|\, \nu_i) = \text{Dirichlet}(\beta^{\nu_i}\nu_i)\,. \tag{8.40}$$

These proposed μ_0 and ν are sampled given the their previous values, and we note that the first μ_0 and ν are computed using the technique discussed in Section 8.5.2. These proposed samples are subsequently used to sample $\mathbf{Q}^{\text{new}}$. We first compute the quantities ς^{new} and κ^{new} using the proposed μ_0^{new} and ν^{new} with Equation (8.19) and Equation (8.20). Then we sample $\mathbf{Q}^{\text{new}}$ given ς^{new} and κ^{new} using the elliptical slice sampler (see Murray *et al.* [2010]):

$$\mathbf{Q}^{\text{new}} \sim \text{GP}(\varsigma^{\text{new}}, \kappa^{\text{new}})\,. \tag{8.41}$$

Finally, we compute the acceptance probability $A' = \min(A, 1)$, where

$$\begin{aligned} A = & \frac{p(\mathbf{Q}^{\text{new}} \,|\, \mathbf{X}, \nu^{\text{new}}, \Xi)}{p(\mathbf{Q}^{\text{old}} \,|\, \mathbf{X}, \nu^{\text{old}}, \Xi)} \frac{f^*(\mu_0^{\text{new}} \,|\, \nu^{\text{new}}, \mathbf{T}) \prod_{i=1}^{A} f^*(\nu_i^{\text{new}} \,|\, \mathbf{T})}{f^*(\mu_0^{\text{old}} \,|\, \nu^{\text{old}}, \mathbf{T}) \prod_{i=1}^{A} f^*(\nu_i^{\text{old}} \,|\, \mathbf{T})} \\ & \times \frac{q(\mu_0^{\text{old}} \,|\, \mu_0^{\text{new}}) \prod_{i=1}^{A} q(\nu_i^{\text{old}} \,|\, \nu_i^{\text{new}})}{q(\mu_0^{\text{new}} \,|\, \mu_0^{\text{old}}) \prod_{i=1}^{A} q(\nu_i^{\text{new}} \,|\, \nu_i^{\text{old}})}\,, \end{aligned} \tag{8.42}$$

and we define $f^*(\mu_0 \,|\, \nu, \mathbf{T})$ and $f^*(\nu \,|\, \mathbf{T})$ as

$$f^*(\mu_0 \,|\, \nu, \mathbf{T}) = \prod_{k=1}^{K} (\mu_{0k})^{t_k^{\mu_1} + \sum_{i=1}^{A} \nu_i}\,, \tag{8.43}$$

$$f^*(\nu_i \,|\, \mathbf{T}) = \prod_{k=1}^{K} (\nu_{ik})^{\sum_{d=1}^{D} t_k^{\eta_d} I(a_d = i)}\,. \tag{8.44}$$

Algorithm 8.2 Performing the MH algorithm for one iteration.

1. Propose a new μ_0^{new} with Equation (8.39).

2. For each author i, propose a new ν_i^{new} with Equation (8.40).

3. Compute the mean function ς^{new} and the covariance matrix κ^{new} with Equation (8.19) and Equation (8.20).

4. Sample $\mathbf{Q}^{\text{new}}$ from Equation (8.41) using the elliptical slice sampler from [Murray *et al.*, 2010].

5. Accept or reject the samples with acceptance probability from Equation (8.42).

The $f^*(\cdot)$ corresponds to the topic model posterior of the variables μ_0 and ν after we represent them as probability vectors explicitly. Note that we treat the acceptance probability A as 1 when the expression in Equation (8.42) evaluates to more than 1. We then accept the proposed samples with probability A, if the sample are not accepted, we keep the respective old values. This completes one iteration of the MH scheme. We summarise the MH algorithm in Algorithm 8.2.

We note that there are some changes to the collapsed blocked Gibbs sampler after we represent μ_0 and ν as probability vectors explicitly. First, as previously mentioned, we do not store their customer counts and the table counts.[43] As a consequence, we stop decrementing the counts at node μ_0 and ν_i. Second, we now sample the topics conditioned on μ_0 and ν. Fortunately, we do not need to re-derive the collapsed blocked Gibbs sampler, we simply replace the modularised likelihood $f(\mu_0)$ and $f(\nu_i)$ in Equation (8.22) by $f^*(\mu_0 \,|\, \nu, \mathbf{T})$ and $f^*(\nu_i \,|\, \mathbf{T})$ respectively and perform the collapsed blocked Gibbs sampling as usual. Lastly, we do not sample the hyperparameters that belong to μ_0 and ν. We discuss the hyperparameter sampler in the next section.

8.5.4 Hyperparameter Sampling

As in previous chapters, we sample the hyperparameters β using an auxiliary variable sampler while leaving α fixed. We note that the auxiliary variable sampler for PYPs that have multiple parents are identical to that of PYPs with single parent, since the sampler only used the total customer counts $C^{\mathcal{N}}$ and the total table counts $T^{\mathcal{N}}$ for a PYP $\mathcal{N}$. Thus we refer the readers to Section 5.4.3 for details.

We would like to point out that hyperparameter sampling is performed for all PYPs in TNTM for the first one thousand iterations. After that, as μ_0 and ν are repre-

[43] However, in the implementation we actually store the customer counts for ν, as they correspond to the summation term in Equation (8.44), though this is only for optimisation reason.

Algorithm 8.3 Full inference algorithm for the TNTM.

1. Initialise the HPYP topic model by assigning random topic to the latent topic z_{dn} associated with each word w_{dn}, and to the latent topic z'_{dm} associated with each hashtag y_{dm}. Then update all the relevant customer counts $\mathbf{C}$ and table counts $\mathbf{T}$ using Equation (8.26) and Equation (8.27), the table counts are initialised equally such that the total table counts are about half of the customer counts.
2. For each word w_{dn} in each document d, perform the following:
    (a) Decrement the counts associated with w_{dn} (see Section 8.5.1.1).
    (b) Blocked sample a new topic for z_{dn} and corresponding customer counts $\mathbf{C}$ and table counts $\mathbf{T}$ (with Equation (8.31)).
    (c) Update (increment counts) the topic model based on the sample.
3. For each hashtag y_{dm} in each document d, perform the following:
    (a) Decrement the counts associated with y_{dm} (see Section 8.5.1.1).
    (b) Blocked sample a new topic for z'_{dn} and corresponding customer counts $\mathbf{C}$ and table counts $\mathbf{T}$ (with Equation (8.32)).
    (c) Update (increment counts) the topic model based on the sample.
4. Sample the hyperparameter $\beta^{\mathcal{N}}$ for each PYP $\mathcal{N}$ (see Section 8.5.4).
5. Repeat Steps 2 – 4 for 1,000 iterations.
6. Alternatingly perform the MH algorithm (Algorithm 8.2) and the collapsed blocked Gibbs sampler conditioned on μ_0 and ν (see Section 8.5.1).
7. Sample the hyperparameter $\beta^{\mathcal{N}}$ for each PYP $\mathcal{N}$ except for μ_0 and ν (see Section 8.5.4).
8. Repeat Steps 6 – 7 until the model converges or when a fix number of iterations is reached.

sented as probability vectors explicitly, we only sample the hyperparameters for the other PYPs (except μ_0 and ν). We note that sampling the concentration parameters allows the topic distributions of each author to vary, that is, some authors have few very specific topics and some other authors can have a wider range of topics. For simplicity, we fix the kernel hyperparameters s, l and σ to 1. Additionally, we also make the priors for the mixing proportions uninformative by setting the λ to 1. We summarise the full inference algorithm for the TNTM in Algorithm 8.3.

Table 8.2: Keywords for querying the datasets in this chapter. The T6 dataset is queried with six hashtags for diversity. While the other three datasets are queried with ten keywords each, as described in Mehrotra *et al.* [2013].

Dataset	Queries
T6	#sport, #music, #finance, #politics, #science and #tech
Generic	business, design, family, food, fun, health, movie, music, space, sport
Specific	Apple, baseball, Burgerking, cricket, France, Mcdonalds, Microsoft, Obama, Sarkozy, United States
Events	attack, conference, Flight 447, Iran election, Jackson, Lakers, recession, scandal, swine flu, T20

8.6 Data

For evaluation of the TNTM, we construct a tweet corpus from the *Twitter 7* dataset [Yang and Leskovec, 2011],[44] This corpus is queried using the hashtags *#sport*, *#music*, *#finance*, *#politics*, *#science* and *#tech*, chosen for diversity. We remove the non-English tweets with *langid.py* [Lui and Baldwin, 2012]. We obtain the data on the followers network from Kwak *et al.* [2010].[45] However, note that this followers network data is not complete and does not contain information for all authors. Thus we filter out the authors that are not part of the followers network data from the tweet corpus. Additionally, we also remove authors who have written less than fifty tweets from the corpus. We name this corpus T6 since it is queried with six hashtags. It is consists of 240,517 tweets with 150 authors after filtering.

Besides the T6 corpus, we also use the tweet datasets described in Mehrotra *et al.* [2013]. The datasets contains three corpora, each of them is queried with exactly ten query terms. The first corpus, named the Generic Dataset, are queried with generic terms. The second is named the Specific Dataset, which is composed of tweets on specific named entities. Lastly, the Events Dataset is associated with certain events. The query terms are presented in Table 8.2. The datasets are mainly used for comparing the performance of the TNTM against the tweet pooling techniques in Mehrotra *et al.* [2013]. We present a summary of the tweet corpora used in this chapter in Table 8.3.

[44] http://snap.stanford.edu/data/twitter7.html (last accessed 10 December 2013)

[45] http://an.kaist.ac.kr/traces/WWW2010.html (last accessed 10 December 2013)

Table 8.3: Summary of the datasets used in this chapter, showing the number of tweets (D), authors (A), unique word tokens ($|\mathcal{V}|$), and the average number of words and hashtags in each tweet. The T6 dataset is queried with six different hashtags and thus has a higher number of hashtags per tweet. The last three datasets are each queried with ten keywords, shown in Table 8.2. We note that there is a typo on the number of tweets for the Events Dataset in Mehrotra *et al.* [2013], the correct number is 107,128.

Dataset	Tweets	Authors	Vocabulary	Words/Tweet	Hashtags/Tweet
T6	240 517	150	5 343	6.35	1.34
Generic	359 478	213 488	14 581	6.84	0.10
Specific	214 580	116 685	15 751	6.31	0.25
Events	107 128	67 388	12 765	5.84	0.17

8.7 Text Preprocessing

We employ a simple preprocessing pipeline for the tweet corpora. We keep two lists for each tweet, one for the observed words and one for the observed hashtags in that tweet. we remove the prefix # from the observed hashtags such that the hashtags share the same token as the words. Additionally, for every seen hashtag in a tweet, we add a copy of the hashtag into the list of words, this is because occasionally the hashtags are used as words in tweets, such as when they are used as part of a sentence.

Next, we perform standard preprocessing techniques such as decapitalising the words and the hashtags, removing stop words, commonly occurred words and rarely occurred words. We also discard the url from the tweets.

Finally, we randomly select 90 % of the dataset as training documents and use the rest for testing. We note that no special consideration is needed in splitting the dataset. The test set can contains unseen hashtags and unseen words, which is unlike Bundschus *et al.* [2009], where the hashtags in the test set need to be seen in training set. We also note that we perform no word normalisation to prevent any loss of meaning of the noisy text.

8.8 Experiments and Results

We consider several tasks to evaluate the TNTM. The first task involves comparing the TNTM with existing baselines on performing topic modelling on tweets. We also compare the TNTM with the random function network model on modelling the followers network. Next, we evaluate the TNTM with ablation studies, in which we perform comparison with the TNTM itself but with each component taken away.

Additionally, we evaluate the clustering performance of the TNTM and topic coherence of the learned topics, we compare the TNTM against the state-of-the-art tweets-pooling LDA method in Mehrotra *et al.* [2013].

8.8.1 Experiment Settings

In all the following experiments, we vary the discount parameters α for the topic distributions μ_0, μ_1, ν_i, η_m, θ'_m, and θ_m, we set α to 0.7 for the word distributions ψ, ϕ' and γ to induce power-law behaviour [Goldwater *et al.*, 2011]. We initialise the concentration parameters β to 0.5, noting that they are learned automatically during inference, we set their hyperprior to $\text{Gamma}(0.1, 0.1)$ for a vague prior. We fix the hyperparameters λ, s, l and σ to 1, as we find that their values have no significant impact on the model performance.[46]

In the following evaluations, we run the full inference algorithm for 2,000 iterations for the models to converge. We note that the MH algorithm only starts after 1,000 iterations, as discussed in Section 8.5.3. We repeat each experiment five times to reduce the estimation error of the evaluation measures.

8.8.2 Goodness-of-fit Test

We compare the TNTM with the HDP-LDA and a nonparametric author-topic model (ATM) on fitting the text data (words and hashtags). Their performances are measured using perplexity on the test set (see Section 5.5.2). However, since tweets are short, we adopt the left to right algorithm [Wallach *et al.*, 2009b] in calculating the test set perplexity, rather than using the document completion method described in Section 5.5.2. In the left to right algorithm, the test set perplexity is computed using a product of conditional probability. The perplexity is

$$\text{Perplexity}(\mathbf{Y}, \mathbf{W}) = \exp\left(- \frac{\log p\big(\mathbf{Y}, \mathbf{W} \,|\, \nu, \mu_1, \psi, \psi'\big)}{\sum_{d=1}^{D} N_d + M_d} \right), \tag{8.45}$$

where the joint likelihood $p\big(\mathbf{W}, \mathbf{Y} \,|\, \nu, \mu_1, \psi, \psi'\big)$ is broken into

$$\begin{aligned} p\big(\mathbf{Y}, \mathbf{W} \,|\, \nu, \mu_1, \psi, \psi'\big) = &\prod_{d=1}^{D} \prod_{m=1}^{M_d} p(y_{dm} \,|\, y_{d1}, \ldots, y_{d,m-1}, \nu, \mu_1, \psi') \\ &\times \prod_{d=1}^{D} \prod_{n=1}^{N_d} p(w_{dn} \,|\, w_{d1}, \ldots, w_{d,n-1}, y_d, \nu, \mu_1, \psi) \,. \end{aligned} \tag{8.46}$$

The conditional probabilities in Equation (8.46) are sequentially evaluated (thus the name left to right), by repeatedly sampling the document–topic distributions given

[46] We vary these hyperparameters over the range of 0.01 to 10 during testing.

Table 8.4: Test perplexity and network log likelihood comparisons between the HDP-LDA, the nonparametric ATM, the random function network model and the TNTM. Lower perplexity indicates better model fitting. The TNTM significantly outperforms the other models in term of model fitting.

Model	Test Perplexity	Network Log Likelihood
HDP-LDA	840.03 ± 15.7	N/A
Nonparametric ATM	664.25 ± 17.76	N/A
Random Function	N/A	−557.86 ± 11.2
TNTM	**505.01** ± 7.8	**−500.63** ± 13.6

the seen words and/or hashtags. For example, the conditional probability for hashtag y_{dm} is evaluated as

$$\begin{aligned} p(y_{dm} \,|\, y_{d1}, \dots, y_{d,m-1}, \nu, \mu_1, \psi') &= \int_{\theta'_d} p(y_{dm} \,|\, \theta'_d, \psi')\, p(\theta'_d \,|\, y_{d1}, \dots, y_{d,m-1}, \nu, \mu_1)\, \mathrm{d}\theta'_d \\ &\approx \frac{1}{R} \sum_{i=1}^{R} p\Big(y_{dm} \,\Big|\, \hat{\theta}'^{(i)}_{d}, \psi'\Big) \\ &= \frac{1}{R} \sum_{i=1}^{R} \sum_{k=1}^{K} \hat{\theta}'^{(i)}_{dk}\, \psi'_{k y_{dm}} \ , \end{aligned} \tag{8.47}$$

where $\hat{\theta}'^{(i)}_{d}$ is a Monte Carlo sample of θ'_d from $p(\theta'_d \,|\, y_{d1}, \dots, y_{d,m-1}, \nu, \mu_1)$. The procedure to estimate the conditional probability for word w_{dn} is similar.

We also compare the TNTM against the original random function network model in terms of the log likelihood of the network data, given by $\log p(\mathbf{X} \,|\, \nu)$. We present the comparison of the perplexity and the network log likelihood in Table 8.4. We note that for the network log likelihood, the less negative the better. From the result, we can see that the TNTM achieves a much lower perplexity compared to the HDP-LDA and the nonparametric ATM. Also, the nonparametric ATM is significantly better than the HDP-LDA. This clearly shows that using more auxiliary information gives a better model fitting. Additionally, we can also see that jointly modelling the text and network data leads to a better modelling on the followers network.

8.8.3 Ablation Test

Next, we perform an extensive ablation study with the TNTM. The components that are tested in this study are (1) authorship, (2) hashtags, (3) PYP μ_1, (4) connection between PYP θ'_d and θ_d, and (5) power-law behaviours on the PYPs. We compare the full TNTM against variations in which each component is ablated (removed).

We discuss in details what happens to the TNTM when a component is ablated. For the TNTM with the authorship component removed, we simply treat all tweets

Table 8.5: Ablation test on the TNTM. The test perplexity and the network log likelihood is evaluated on the TNTM against several ablated variants of the TNTM. Again, lower perplexity means better model fitting. The result shows that each component in the TNTM is important.

TNTM Model	Test Perplexity	Network Log Likelihood
No author	669.12 ± 9.3	N/A
No hashtag	1017.23 ± 27.5	-522.83 ± 17.7
No μ_1 node	607.70 ± 10.7	-508.59 ± 9.8
No θ'-θ connection	551.78 ± 16.0	-509.21 ± 18.7
No power-law	508.64 ± 7.1	-560.28 ± 30.7
Full model	$\mathbf{505.01} \pm 7.8$	$-\mathbf{500.63} \pm 13.6$

as being written by a single author, that is, $A = 1$. This essentially changes ν into a layer of PYP prior for the document–topic distributions η and removes the network component. For the TNTM without hashtags, the hashtags are treated as part of the words, that is, they are now captured by the variables $\mathbf{W}$. In this ablated model, the hashtags retain the # prefix, allowing us to distinguish the hashtags from words.

On the other hand, the TNTM with PYP μ_1 removed and the TNTM with the connection between θ'_d and θ_d removed are simply just the respective models without such components. For example, when μ_1 is removed, θ'_d now has a single parent, which is η_d. This is similar for the case when the connection between θ'_d and θ is dropped, where now the θ has only one parent. Note that implementing these models is straightforward with the flexible framework discussed in Chapter 5, since we would only need to re-specify the models. Finally, removing the power-law behaviours in the PYPs is as simple as setting the discount parameter α to 0.

Table 8.5 presents the test set perplexity and the network log likelihood of these models, it shows significant improvements of the TNTM over the ablated models. From this, we see that the greatest improvement on perplexity is from modelling the hashtags, which suggests that the hashtag information is the most important for modelling tweets. Second to the hashtags, the authorship information is very important as well. On the contrary, the improvement on perplexity is marginal when we model the power-law behaviour in the PYPs, this implies that the words in tweets do not necessarily exhibit a power-law behaviour, which is reasonable since the tweets are too short. However, even though modelling the power-law behaviour is not that important for perplexity, we see that the improvement on the network log likelihood is best achieved by modelling the power-law. This is because the flexibility enables us to learn the author–topic distributions better, and thus allowing the TNTM to fit the network data better. This also suggests that the authors in the corpus tend to focus on a specific topic rather than having a wide interest.

8.8.4 Document Clustering and Topic Coherence

Mehrotra *et al.* [2013] shows that running LDA on pooled tweets rather than unpooled tweets gives significant improvement on clustering evaluation and topic coherence. In particular, they find that grouping tweets based on the hashtags provides most improvement. In this section, we show that instead of resorting to such *ad-hoc* method, the TNTM can achieve a significantly better performance. The clustering evaluations are measured with purity and normalised mutual information (NMI, see Section 5.5.4) , and topic coherence is measured by pointwise mutual information (PMI) [Newman *et al.*, 2009]. Since the ground truth labels are unknown, we use the respective query terms as the ground truth labels for evaluations. Note that tweets that satisfy multiple labels are removed. Given the learned model, we assign a tweet to a cluster based on its dominant topic:

$$\text{Dominant Topic}(d) = \arg\max_k \eta_{dk} \,. \tag{8.48}$$

Next, we follows Mehrotra *et al.* [2013] and use the top ten words in each topic for the computation of PMI. Denoting the top ten words of each topic k as $\omega_{k1}, \ldots, \omega_{k10}$, the PMI is calculated as

$$\text{PMI}(\omega) = \frac{1}{100K} \sum_{k=1}^{K} \sum_{i=1}^{10} \sum_{j=1}^{10} \text{Score}(\omega_{ki}, \omega_{kj}) \,, \tag{8.49}$$

where the score for each pair of word is given as

$$\text{Score}(\omega_{ki}, \omega_{kj}) = \log \frac{\hat{p}(\omega_{ki}, \omega_{kj})}{\hat{p}(\omega_{ki})\, \hat{p}(\omega_{kj})} \,. \tag{8.50}$$

The $\hat{p}(\cdot)$ is the empirical frequency of a word occurring in the tweet corpus, while $\hat{p}(\cdot,\cdot)$ is the empirical frequency of a pair of words co-occurring.

We perform the evaluations on the Generic, Specific and Events datasets for comparison purpose. We note the lack of network information in these datasets, and thus we employ only the HPYP part of the TNTM. Additionally, since the purity can trivially be improved by increasing the number of clusters, we limit the maximum number of topics to twenty for a fair comparison. We present the results in Table 8.6 and Table 8.7. We can see that the TNTM outperforms the pooling method in all aspects except on the Specific dataset, where it achieves the same purity as the best pooling scheme, but worse PMI score compared to two of the pooling methods.

Table 8.6: Clustering evaluations of the TNTM against the LDA with different pooling schemes. Note that higher purity and NMI indicate better performance. The results for the different pooling methods are obtained from Table 4 in Mehrotra *et al.* [2013]. The TNTM achieves better performance on the purity and the NMI for all datasets except for the Specific dataset, where it obtains the same purity score as the best pooling method.

Method/Model	Purity			NMI		
	Generic	**Specific**	**Events**	**Generic**	**Specific**	**Events**
No pooling	0.49	0.64	0.69	0.28	0.22	0.39
Author	0.54	0.62	0.60	0.24	0.17	0.41
Hourly	0.45	0.61	0.61	0.07	0.09	0.32
Burstwise	0.42	0.60	0.64	0.18	0.16	0.33
Hashtag	0.54	**0.68**	0.71	0.28	0.23	0.42
TNTM	**0.66**	**0.68**	**0.79**	**0.43**	**0.31**	**0.52**

Table 8.7: Comparison of the topic coherence for the topics learned by the TNTM against the LDA with different pooling methods. Here, the more positive the PMI, the better perceived the learned topics are. The results for the different pooling methods are obtained from Table 4 in Mehrotra *et al.* [2013]. The topics learned by the TNTM have greater coherence compared to the pooling methods, with the exception on the Specific dataset, where the corresponding PMI is lower than two of the pooling schemes.

Method/Model	PMI		
	Generic	**Specific**	**Events**
No pooling	−1.27	0.47	0.47
Author	0.21	0.79	0.51
Hourly	−1.31	0.87	0.22
Burstwise	0.48	0.74	0.58
Hashtag	0.78	**1.43**	1.07
TNTM	**0.79**	0.81	**1.66**

8.9 Qualitative Analysis of Learned Topic Models

Here, we present some qualitative analysis of the learned TNTM. We first inspect the learned topic–word distributions, and we propose a way to perform automatic topic labelling using hashtags. In addition, we analyse the author–topic distributions, we find that the using the topic–hashtag distributions allows us to understand the authors better.

Table 8.8: Topical analysis of the learned TNTM on the T6 dataset, which displays the top three hashtags and the top n words on six topics. Instead of manually assigning a topic label to the topics, we find that the top hashtags can serve as the topic labels for the topics.

Topic	Top Hashtags	Top Words
Topic 1	finance, money, economy	finance, money, bank, marketwatch, stocks, china, group, shares, sales
Topic 2	politics, iranelection, tcot	politics, iran, iranelection, tcot, tlot, topprog, obama, musiceanewsfeed
Topic 3	music, folk, pop	music, folk, monster, head, pop, free, indie, album, gratuit, dernier
Topic 4	sports, women, asheville	sports, women, football, win, game, top, world, asheville, vols, team
Topic 5	tech, news, jobs	tech, news, jquery, jobs, hiring, gizmos, google, reuters
Topic 6	science, news, biology	science, news, source, study, scientists, cancer, researchers, brain, biology, health

8.9.1 Automatic Topic Labelling

There have been recent attempts to label topics automatically in topic modelling. For instance, Lau *et al.* [2011] use Wikipedia to extract labels for topics, and Mehdad *et al.* [2013] use the entailment relations to select relevant phrases for topics. Recall that in the previous chapters, we have manually assign a topic for each topic–word distribution. Here, we show that using the hashtags allows us to get good labels for the topics.

In Table 8.8, we display the top words from the topic–word distribution ψ_k for each topic k, instead of manually assigning the topic labels, we display the top three hashtags from the topic–hashtag distribution ψ'_k. As we can see from Table 8.8, the hashtags appear suitable to be used as topic labels. In fact, by empirically evaluating the suitability of the hashtags in representing the topics, we consistently find that, over 90 % of the hashtags are good candidates for the topic labels. Moreover, inspecting the topics show that the major hashtags coincide with the query terms used in constructing the T6 dataset, which is to be expected. This indirectly verifies that the TNTM is working properly.

8.9.2 Analysing the Authors' Topics

Next, we move on to analyse the topic areas for the authors in the T6 dataset, by inspecting the learned author–topic distributions ν from the TNTM. We look at the

Table 8.9: Inference on authors' interest. The authors' topics are represented by the top hashtags from the topic–hashtag distributions ψ'. From the result, there is an obvious relationship between the authors' Twitter ID and their topics.

Twitter ID	Dominant Topic
finance_yard	finance, money, realestate
ultimate_music	music, ultimatemusiclist, mp3
seriouslytech	technology, web, tech
seriouspolitics	politics, postrank, news
pr_science	science, news, postrank

authors in which their Twitter IDs convey the topics they are interested in. We then determine their dominant topic from the author–topic distributions ν :

$$\text{Dominant Topic}(i) = \arg\max_k \, \nu_{ik} \, , \tag{8.51}$$

We display the top hashtags corresponding to the authors' dominant topic in Table 8.9. We find that there is matching between the authors and their topics illustrated by hashtags. For example, the topic for the author finance_yard is represented by #finance, #money and #realestate, this points out that the author mainly tweets about finance related posts, which is unsurprising given that the Twitter ID contains the word *finance*.

8.10 Diagnostics

In this section, we perform simple diagnostic checks to assess the learning algorithm of the TNTM. In particular, we assess the convergence of the MCMC algorithm and examine the mixing probabilities correspond to the PYPs with multiple parents.

8.10.1 Convergence Analysis of the MCMC Algorithms

We assess the convergence of the inference algorithm of the TNTM. In Figure 8.2, we display the training log likelihoods, $\log p(\mathbf{W}, \mathbf{Y} \,|\, \mathbf{Z}, \mathbf{Z}', \psi, \psi')$, for the TNTM and the corresponding ablated models. We also show the log likelihood for the TNTM without running the MH algorithm on the network.

In Table 8.2, we can see that the full TNTM model converges much faster than the other models, it also achieves a higher training log likelihood. Additionally, we would like to point out that the log likelihood improves further when we start running the MH algorithm after the first 1,000 iterations, though the improvement is not realised immediately after starting the MH algorithm. This indicates that modelling

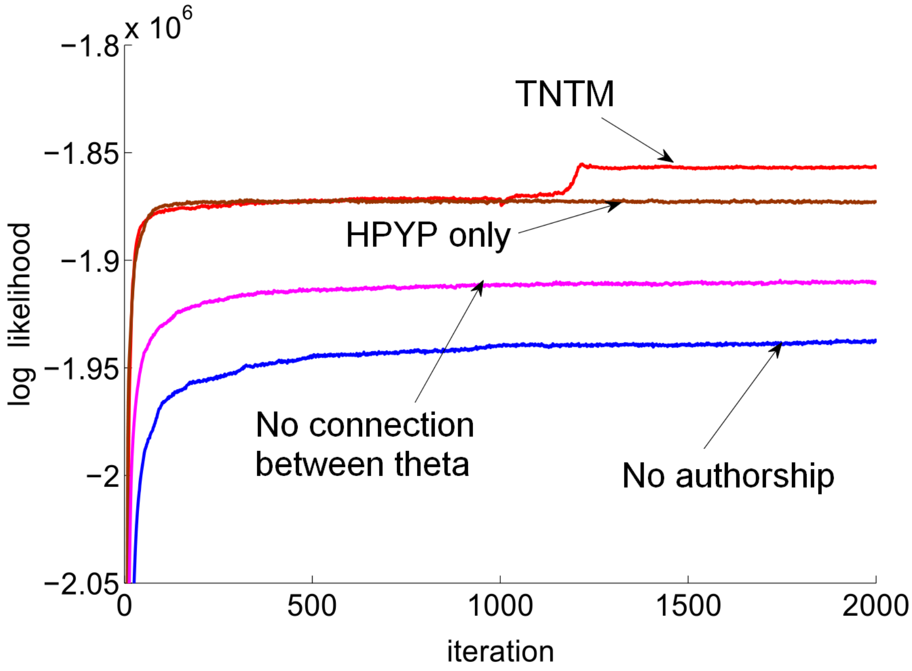


Figure 8.2: Convergence analysis for the TNTM and the ablated TNTM models. The full model achieves the best training log likelihood and also with the shortest amount of iterations. The purple line shows the convergence for the TNTM but with the connection between θ'_d and θ_d cut off. The blue line shows the convergence for the TNTM but without using the authorship information. By comparing the full inference algorithm of the TNTM with the collapsed Gibbs sampler of its HPYP model, we can see that running the MH algorithm on the followers network leads to further improvement on the training log likelihood.

the followers network leads to better modelling on the text. We note that not all ablated models are shown here because some of the models converge to a lower log likelihood, and thus removed for presentation reason. Another interesting observation in the experiments is that the acceptance rate of the MH algorithm is relatively high (average 56 %), indicating a good proposal distribution in the MH sampler.

8.10.2 Inspecting the Mixing Proportions of the PYPs

The posterior of the mixing proportion $\rho^{\mathcal{N}}$ of a PYP $\mathcal{N}$ gives us insight on the influence of its parent PYPs. To illustrate, the mixing proportion $\rho^{\theta'_d}$ is the proportion of the influence of the miscellaneous topic distribution μ_1 to θ'_d, and $1 - \rho^{\theta'_d}$ is the proportion of influence from the author–topic distributions ν. We estimate the mixing proportions with their posterior mean. The procedure to compute the posterior mean is outlined in Section 8.5.2.

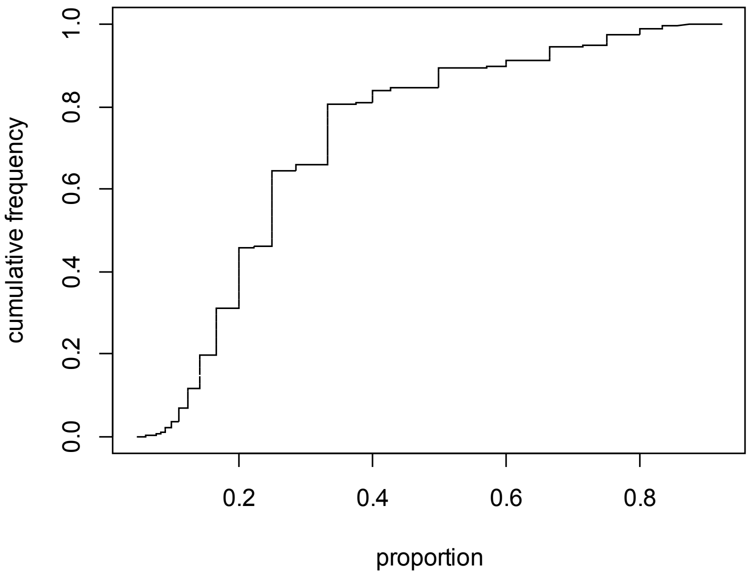


Figure 8.3: Cumulative frequency of the mixing proportions $\rho^{\theta'_d}$. The plot shows that more than half of the mixing proportions are smaller than 0.3, and more than 80 % of the mixing proportions are smaller than 0.4. This indicates that η_d has stronger influence to θ'_d compared to μ_1.

In the TNTM, we expect the $\rho^{\theta'_d}$ to be small since the miscellaneous topic distribution μ_1 is designed to capture topics that are not frequently used by authors. To confirm this, we inspect the mixing proportion $\rho^{\theta'_d}$ in each tweet d, and display their empirical cumulative frequency plot in Figure 8.3. From this figure, we can see that more than 80 % of the estimated $\rho^{\theta'_d}$ are less than 0.4, indicating that the TNTM is working as intended.

Next, we look at the mixing proportion ρ^{θ_d}, which is inversely related to the influence of the θ'_d to θ_d. In Figure 8.4, we similarly plot the empirical cumulative frequency of the mixing proportions ρ^{θ_d} for each tweet d. Although the cumulative frequency curve appears more linear, we can see that more than half of the mixing proportions are greater than 0.6, which shows that θ'_d does influence the θ_d to a certain extent. This suggests that the hashtags is important in the topic modelling of the words.

8.11 Summary

In this chapter, we propose the TNTM, which is a fully Bayesian nonparametric topic model that jointly models tweets and the associated followers network information. The TNTM employs a nonparametric Bayesian approach by using the PYP and the GP, and achieves a flexible modelling by performing inference on a network of PYPs.

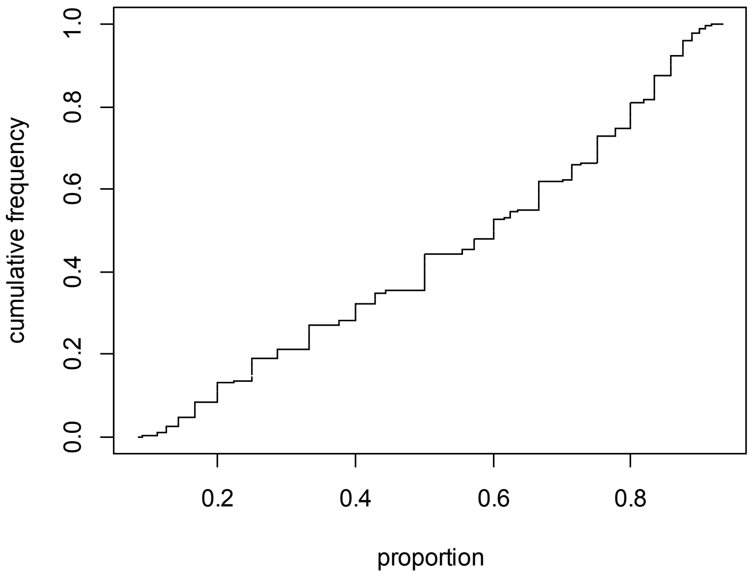


Figure 8.4: Cumulative frequency of the mixing proportions ρ^{θ_d}. In contrast to Figure 8.3, more than half of the mixing proportions are greater than 0.6, and more than 60 % of the mixing proportions are bigger than 0.5. This suggests that the topic proportion for hashtags (θ'_d) influences θ_d more than η_d.

Experiments with Twitter datasets show that the TNTM achieves significant improvement compared to existing baselines. Furthermore, the ablation study demonstrates the usefulness of each component in the TNTM. The TNTM also provides us additional dimensions to analyse the tweet corpora, for instance, using the hashtags as topic labels during topic exploration, and understanding the authors better with the assessment on the author–topic distributions.

Future work on this includes speeding up the posterior inference algorithm, especially for the network model. The MH algorithm requires a cubic time operation (due to matrix operations) and thus does not scale to large network. Potential solutions to this include employing approximation techniques such as by introducing inducing variables [Hensman *et al.*, 2013]. Other future studies would be to incorporate other auxiliary information that is available in social media, such as *location*, *hyperlinks* and *multimedia contents*. It is also interesting to apply the TNTM to other types of data such as blog posts and news feed.

Chapter 9

Conclusion

In this dissertation, we have presented three novel topic models built on the hierarchical Pitman-Yor process (HPYP) using the state-of-the-art Bayesian techniques. These topic models were designed to tailor various text corpora, such as social media messages and scientific papers, that are accompanied by different types of auxiliary information (*e.g.*, geographic location associated with the message, or the primary author who has written the research paper).

9.1 Contributions

Although the proposed topic models are quite different, they share the same principle of using the HPYP to pass information. This allows us to propose a single framework, discussed in Chapter 5, to implement these topic models, where we modularise the PYPs (and other variables) into blocks that can be combined to form different models. Doing so enables significant time to be saved on implementation of the topic models, and it produces a cleaner code. In Chapter 5, we presented a general HPYP topic model, that can be seen as a generalisation to the HDP-LDA [Teh and Jordan, 2010]. The HPYP topic model is represented using a Chinese Restaurant Process (CRP) metaphor [Teh and Jordan, 2010; Blei *et al.*, 2010; Chen *et al.*, 2011], and we discussed how the posterior likelihood of the HPYP topic model can be modularised. We then detailed the learning algorithm for the topic model, taking advantage of the CRP representation and the modularised form. We closed Chapter 5 with a technical discussion on the implementation of the topic model.

As mentioned above, the proposed topic models look at text data with different types of auxiliary information. Our first topic model, named the Twitter Opinion Topic Model (TOTM), explores tweet corpora for sentiment analysis. The TOTM extracts sentiment indicators from tweets and uses external sentiment lexicon to this end. Next, our proposed Citation Network Topic Model (CNTM) performs bibliographic analysis on research publications. The CNTM makes use of the accompanying metadata such as author details. In addition, the CNTM also models the citation network between the publications. Lastly, we presented the Twitter Network Topic model (TNTM) as a fully Bayesian topic model on tweets. The TNTM models the authors, text, hashtags, and the authors-follower network. Our experiments on these different text corpora have suggested that incorporating more auxiliary information into topic models leads to better fitting models, in addition to enabling more ways to visualise the text corpora.

We will now review the proposed topic models in turn, starting with the TOTM introduced in Chapter 6. For sentiment analysis, the TOTM leverages hashtags, mentions, emoticons, and strong sentiment words that are present in tweets. The novelty of the TOTM lies in the modelling of the target-opinion interaction directly, allowing the discovery of target-specific opinions and leading to additional dimensions in visualising the tweet corpora. The TOTM also utilises sentiment lexicons to incorporate sentiment prior information into topic models. It employs a novel formulation that learns and updates with the data. In the experiments, we achieved improvement on model fitting and sentiment classification by using the TOTM over some baselines. For instance, the TOTM is less perplexed by the opinion words in the test set, compared to other models. On a tweet corpus consists of electronic products, we presented additional dimensions to visualise the corpus. Example includes (1) inspecting the sentiment-induced opinions for the targets (products and services), (2) comparing the opinions on brands and products, and (3) looking at the contrastive opinions on certain products. All in all, using these auxiliary information (with TOTM) have allowed us to extract valuable information from noisy platform such as Twitter.

On the other hand, as discussed in Chapter 7, the CNTM is designed for the research publication data for bibliographic analysis. Besides accompanying information such as authors, the CNTM also considers the network between publications through their citations. The novelty of this work comes from a novel and efficient algorithm that allows the learning of the CNTM to mimic the learning algorithm of a generic HPYP topic model mentioned in Chapter 5. This algorithm exploits the posterior of the Poisson distributions and uses approximation to absorb the network component into the topic model component. Furthermore, we proposed a method to incorporate supervision into the CNTM, using the categorical labels of the publications. In the experiments, we showed that our proposed topic models achieve better model fitting and better document clustering compared to two baselines. Moreover, on three publication corpora extracted from CiteSeer$^{\mathrm{X}}$, the topical summary suggests a good clustering of the publication data into various disciplines. The CNTM also allows us to analyse the authors' major research area and lets us visualise the network of the authors through their topics.

Finally, the TNTM is presented in Chapter 8, which is designed for modelling text content, hashtags, authors, and the followers network on tweets. In addition to HPYP, the TNTM employs the Gaussian process (GP) for the network modelling. The main usage of the TNTM is for content discovery on social networks, that is, generic topic modelling. Through experiments, we show that jointly modelling of the text content and the network leads to better model fitting as compared to modelling them separately. Additionally, we also demonstrated that the TNTM outperforms the state-of-the-art tweet pooling method on document clustering and topic coherence. Results on the qualitative analysis show that the learned topics and the authors' topics are sound. Moreover, we found that the hashtags can serve as useful labels for the topics. Note that in all the proposed models, we performed diagnostic checks to assess the learning algorithms. We found that the learning algorithms converge within 2,000 iterations and that the proposed Metropolis-Hastings (MH) algorithms have relatively high acceptance probabilities.

We conclude that auxiliary information can serve as a valuable information to be used in topic models. However, care must be taken to design the appropriate topic models to make use of these information. In this dissertation, we had proposed three topic models using the state-of-the-art Bayesian technologies for various types of text data, which hopefully serve as a reference for future models. In the next section, we outline several avenues for future work.

9.2 Future Research

As future work, it would be interesting to apply the proposed topic models on other types of data. For instance, we can apply the TOTM directly to short reviews without modifying the model.[47] Similarly, we can apply the TNTM to other types of data, such as blogs and news feed. Alternatively, we can also use the proposed models for other applications that were not mentioned, such as hashtags recommendation and content suggestion for new Twitter user. On Twitter, one interesting direction would be to investigate the plausibility of constraining the tweets to have only one topic.

Another line of future studies involves extending the existing topic models to incorporate more auxiliary information in their modelling. To give some examples, we can model the location of the tweets and the embedded multimedia contents such as URL, images and videos. Another interesting kind of information would be the path of a retweeted content.[48] For research publications, additional auxiliary information that can be important includes the time of publication, the publication type, and the conference venue. It is also important to consider utilising more external resources for topic modelling. For instance, besides the sentiment lexicon, we can also make use of synonym and antonym lexicons for sentiment analysis. This also includes using external technologies for cleaning the data, such as applying spam filtering [Tsur *et al.*, 2010; McCord and Chuah, 2011] on tweets.

In topic modelling literature, models are usually designed for a particular type of

[47] For reviews that have explicit ratings, we can replace the emotion indicators in the TOTM by the ratings, though extending the TOTM to model the ratings would also be possible.

[48] A Retweet is a reposting of someone else's tweet.

text corpus. However, it would be interesting to see topic models that are designed for corpora with multiple types of text data. An important application would be to extract political sentiment from various sources like social media, discussion forums and published articles. Another interesting future research would be on combining different kinds of topic models for a complete analysis. One implication of such research would allow us to transfer the learned knowledge from a topic model to another. The work on combining LDA has already been looked at by Schnober and Gurevych [2015]. However, combining other kind of topic models, especially those of nonparametric nature, is largely unexplored.

Last but not least, an important future investigation should be about speeding up the learning algorithm for nonparametric Bayesian topic models by exploring more efficient learning algorithm. Within the same Markov chain Monte Carlo (MCMC) framework, it would be interesting to employ the reversible jump MCMC technique [Green and Hastie, 2009], such as the split-merge MCMC sampler [Jain and Neal, 2004; Wang and Blei, 2012], to hasten the convergence time. Moreover, we can also consider recent work [Li *et al.*, 2014] that utilises the Metropolis-Hastings-Walker sampler to speed up Bayesian inference. Alternatively, other approximate inference techniques are also of interest. For example, employing the variational inference [Blei and Jordan, 2006], the expectation propagation [Minka, 2001], and the expectation maximisation algorithm [Dempster *et al.*, 1977; Moon, 1996] might lead to a faster learning algorithm.

Appendix A

Appendix

A.1 Derivation of Gradient Ascent Algorithm for Hyperparameter Optimisation

We would like to optimise for the hyperparameter b by updating b to its maximum *a posteriori* estimate.

The posterior distribution of b is given by

$$p(b \mid c) \propto p(b) \prod_{r=-1}^{1} \prod_{v=1}^{|\mathcal{V}_o|} (\phi^*_{rv})^{c_{rv}} = p(b) \prod_{r=-1}^{1} \prod_{v=1}^{|\mathcal{V}_o|} \left(\frac{(1+b)^{X_{rv}}}{\sum_i (1+b)^{X_{ri}}} \right)^{c_{rv}} \quad , \tag{A.1}$$

where c_{rv} is the number of times a word v is assigned to sentiment r, and $p(b)$ is the hyperprior of b. We assume a weak hyperprior for b:

$$\begin{aligned} b &\sim \text{Gamma}(1, 1) \quad , \\ p(b) &\propto e^{-b} \quad . \end{aligned} \tag{A.2}$$

Optimising for the posterior is the same as optimising for the log posterior:

$$\begin{aligned} l(b) :=& \log p(b \mid c) \\ =& \sum_{r=-1}^{1} \sum_{v=1}^{|\mathcal{V}_o|} c_{rv} \log \left(\frac{(1+b)^{X_{rv}}}{\sum_i (1+b)^{X_{ri}}} \right) + \log p(b) + \text{constant} \\ =& \sum_{r=-1}^{1} \sum_{v=1}^{|\mathcal{V}_o|} c_{rv} \left(X_{rv} \log(1+b) - \log \left(\sum_i (1+b)^{X_{ri}} \right) \right) + \log p(b) + \text{constant} \quad . \end{aligned} \tag{A.3}$$

We derive the gradient of $l(b)$, denoted as $l'(b)$, as follows:

$$\begin{aligned} l'(b) &= \frac{\mathrm{d}l(b)}{\mathrm{d}b} \\ &= \sum_{r=-1}^{1}\sum_{v=1}^{|\mathcal{V}_o|} c_{rv}\left(\frac{X_{rv}}{1+b} - \frac{\sum_j X_{rj}(1+b)^{X_{rj}-1}}{\sum_i (1+b)^{X_{ri}}}\right) + \rho'(b) \\ &= \frac{1}{1+b}\sum_{r=-1}^{1}\sum_{v=1}^{|\mathcal{V}_o|} c_{rv}\left(X_{rv} - \sum_j X_{rj}\frac{(1+b)^{X_{rj}}}{\sum_i (1+b)^{X_{ri}}}\right) + \rho'(b) \\ &= \frac{1}{1+b}\sum_{r=-1}^{1}\sum_{v=1}^{|\mathcal{V}_o|} c_{rv}\big(X_{rv} - \mathbb{E}_{\phi_r^*}[X_r]\big) + \rho'(b) \quad , \end{aligned} \tag{A.4}$$

where $\rho'(b)$ is defined as the derivative of the log prior of b, $\frac{\mathrm{d}\log p(b)}{\mathrm{d}b}$, and $\mathbb{E}_{\phi_r^*}[X_r]$ is the expected score of sentiment r under the probability distribution ϕ_r^*:

$$\mathbb{E}_{\phi_r^*}[X_r] = \sum_j X_{rj}\,\phi_{rj}^* \quad . \tag{A.5}$$

In addition, we can also derive the second derivative $l''(b)$:

$$l''(b) = -(1+b)^{-2}\sum_{r=-1}^{1}\sum_{v=1}^{|\mathcal{V}_o|} c_{rv}\big(X_{rv} + \mathbb{V}_{\phi_r^*}[X_r] - \mathbb{E}_{\phi_r^*}[X_r]\big) + \rho''(b) \quad , \tag{A.6}$$

where $\mathbb{V}_{\phi_r^*}[X_r]$ is the variance of X_r under ϕ_r^*. The second derivative can be used to verify that the optimal value obtained from the gradient ascent algorithm corresponds to the maxima.

A.2 Delta Method Approximation

We employ the delta method to show that

$$\int f(\theta)\,\exp\big(-g(\theta)\big)\,\mathrm{d}\theta \approx \exp\Big(-g(\hat{\theta})\Big)\int f(\theta)\,\mathrm{d}\theta \qquad \text{for small } g(\hat{\theta}) \quad , \tag{A.7}$$

where $\hat{\theta}$ is the expected value according to a distribution proportional to $f(\theta)$, more specifically, define $p(\theta)$ as the probability density of θ, we have

$$\hat{\theta} = \mathbb{E}[\theta] = \int \theta\, p(\theta)\,\mathrm{d}\theta \quad , \qquad\qquad f(\theta) = \text{constant} \times p(\theta) \quad . \tag{A.8}$$

First we note that the Taylor expansion for a function $h(\theta) = \exp\big(-g(\theta)\big)$ at $\hat{\theta}$ is

$$h(\theta) = \sum_{n=0}^{\infty}\frac{1}{n!}\Big(h^{(n)}(\hat{\theta})\Big)\big(\theta-\hat{\theta}\big)^n \quad , \tag{A.9}$$

where $h^{(n)}\big(\hat{\theta}\big)$ denotes the n-th derivative of $h(\cdot)$ evaluated at $\hat{\theta}$:

$$h^{(n)}\big(\hat{\theta}\big) = \Big(- g'\big(\hat{\theta}\big)\Big)^n \, h\big(\hat{\theta}\big) \quad . \tag{A.10}$$

Multiply Equation A.9 with $f(\theta)$ and integrating gives

$$\begin{aligned} \int f(\theta)\, h(\theta)\, \mathrm{d}\theta &= \sum_{n=0}^{\infty} \frac{1}{n!} \Big(h^{(n)}\big(\hat{\theta}\big)\Big) \int f(\theta)\, \big(\theta - \hat{\theta}\big)^n \,\mathrm{d}\theta \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \Big(- g'\big(\hat{\theta}\big)\Big)^n \int f(\theta)\, \big(\theta - \hat{\theta}\big)^n \,\mathrm{d}\theta \quad . \end{aligned} \tag{A.11}$$

Since $g\big(\hat{\theta}\big)$ is small, the term $\big(-g'\big(\hat{\theta}\big)\big)^n$ becomes exponentially smaller as n increases. Here we let $\big(-g'\big(\hat{\theta}\big)\big)^n \approx 0$ for $n \geq 2$. Hence, continuing from Equation A.11:

$$\begin{aligned} \int f(\theta)\, h(\theta)\, \mathrm{d}\theta \;&\approx\; h(\hat{\theta}) \int f(\theta)\, \mathrm{d}\theta + \Big(- g'\big(\hat{\theta}\big)\Big) h(\hat{\theta}) \; \underbrace{\int f(\theta)\, \big(\theta - \hat{\theta}\big)\, \mathrm{d}\theta}_{0} \\ &\approx\; h(\hat{\theta}) \int f(\theta)\, \mathrm{d}\theta \quad . \end{aligned} \tag{A.12}$$

A.3 Keywords for Querying CiteSeer Datasets

1. For ML dataset:

Machine Learning: *machine learning, neural network, pattern recognition, indexing term, support vector machine, learning algorithm, computer vision, face recognition, feature extraction, image processing, high dimensionality, image segmentation, pattern classification, real time, feature space, decision tree, principal component analysis, feature selection, backpropagation, edge detection, object recognition, maximum likelihood, statistical learning theory, supervised learning, reinforcement learning, radial basis function, support vector, em algorithm, self organization, image analysis, hidden markov model, artificial neural network, independent component analysis, genetic algorithm, statistical model, dimensional reduction, indexation, unsupervised learning, gradient descent, large scale, maximum likelihood estimate, statistical pattern recognition, cluster algorithm, markov random field, error rate, optimization problem, satisfiability, high dimensional data, mobile robot, nearest neighbour, image sequence, neural net, speech recognition, classification accuracy, diginal image processing, factor analysis, wavelet transform, local minima, probability distribution, back propagation, parameter estimation, probabilistic model, feature vector, face detection, objective function, signal processing, degree of freedom, scene analysis, efficient algorithm, computer simulation, facial expression, learning problem, machine vision, dynamic system, bayesian network, mutual information, missing value, image database, character recognition, dynamic program,* *finite mixture model, linear discriminate analysis, image retrieval, incomplete data, kernel method, image representation, computational complexity, texture feature, learning method, prior knowledge, expectation maximization, cost function, multi layer perceptron, iterated reweighted least square, data mining.*

2. For M10 dataset:

Biology: *enzyme, gene expression, amino acid, escherichia coli, transcription factor, nucleotides, dna sequence, saccharomyces cerevisiae, plasma membrane, embryonics.*

Computer Science: *neural network, genetic algorithm, machine learning, information retrieval, data mining, computer vision, artificial intelligent, optimization problem, support vector machine, feature selection.*

Social Science: *developing country, higher education, decision making, health care, high school, social capital, social science, public health, public policy, social support.*

Financial Economics: *stock returns, interest rate, stock market, stock price, exchange rate, asset prices, capital market, financial market, option pricing, cash flow.*

Material Science: *microstructures, mechanical property, grain boundary, transmission electron microscopy, composite material, materials science, titanium, silica, differential scanning calorimetry, tensile properties.*

Physics: *magnetic field, quantum mechanics, field theory, black hole, kinetics, string theory, elementary particles, quantum field theory, space time, star formation.*

Petroleum Chemistry: *fly ash, diesel fuel, methane, methyl ester, diesel engine, natural gas, pulverised coal, crude oil, fluidised bed, activated carbon.*

Industrial Engineering: *power system, construction industry, induction motor, power converter, control system, voltage source inverter, permanent magnet, digital signal processor, sensorless control, field oriented control.*

Archaeology: *radiocarbon dating, iron age, bronze age, late pleistocene, middle stone age, upper paleolithic, ancient dna, early holocene, human evolution, late holocene.*

Agriculture: *irrigation water, soil water, water stress, drip irrigation, grain yield, crop yield, growing season, soil profile, soil salinity, crop production*

3. For AvS dataset:

History: *nineteeth century, cold war, south africa, foreign policy, civil war, world war ii, latin america, western europe, vietnam, middle east.*

Religion: *social support, foster care, child welfare, human nature, early intervention, gender difference, sexual abuse, young adult, self esteem, social services.*

Physics: *magnetic field, quantum mechanics, string theory, field theory, numerical simulation, black hole, thermodynamics, phase transition, electric field, gauge theory.*

Chemistry: *crystal structure, mass spectrometry, copper, aqueous solution, binding site, hydrogen bond, oxidant stress, free radical, liquid chromatography, organic compound.*

Biology: *genetics, enzyme, gene expression, polymorphism, nucleotides, dna sequence, saccharomyces cerevisiae, cell cycle, plasma membrane, embryonics.*

A.4 Recovering Word Counts from TF-IDF

The PubMed dataset [Sen *et al.*, 2008] was preprocessed to TF-IDF (term frequency-inverse document frequency) format, that is, the raw word count information is lost. Here, we describe how we can recover the word count information, using a simple and reasonable assumption — that the least occurring words in a document occur only once.

We denote t_{dw} as the TF-IDF for word w in document d, f_{dw} as the corresponding term frequency (TF), and i_w as the inverse document frequency (IDF) for word w. Our aim is to recover the word counts c_{dw} given the TF-IDF. The TF-IDF is computed[49] as

$$t_{dw} = f_{dw} \times i_w \ , \qquad f_{dw} = \frac{c_{dw}}{\sum_w c_{dw}} \ , \qquad i_w = \log \frac{\sum_d 1}{\sum_d I(c_{dw} > 0)} \ , \tag{A.13}$$

where $I(\cdot)$ is the indicator function.

We note that $I(c_{dw} > 0) = I(t_{dw} > 0)$ since the TF-IDF for a word w is positive if and only if the corresponding word count is positive. This allows us to compute the IDF i_w easily from Equation A.13. We can then determine the TF:

$$\begin{aligned} f_{dw} &= t_{dw} / i_w \\ &= t_{dw} \times \left(\log \frac{\sum_d 1}{\sum_d I(t_{dw} > 0)} \right)^{-1} \ . \end{aligned} \tag{A.14}$$

Now we are left with computing c_{dw} given the f_{dw}, however, we can obtain infinitely many solutions since we can always multiply c_{dw} by a constant and get the same f_{dw}. Luckily, since we are working with natural language, it is reasonable to assume that the least occurring words in a document occur only once, mathematically, this is given as

$$c_{dw} = 1 \qquad \text{for } \ w = \arg\min_w f_{dw} \ . \tag{A.15}$$

Thus we can work out the normaliser $\sum_w c_{dw}$ and recover the word counts for all words in all documents.

$$\sum_w c_{dw} = \frac{1}{\min_w f_{dw}} \ , \qquad\qquad c_{dw} = f_{dw} \times \sum_w c_{dw} \ . \tag{A.16}$$

[49] Note that there are multiple ways to define a TF-IDF in practice. The specific TF-IDF formula used by the PubMed dataset was determined *via* trial-and-error and elimination.

A.5 Exclusion Words to Detect Incorrect Authors

A list of words we use to filter out invalid authors during the preprocessing step:

society, university, universität, universitat, author, advisor, acknowledgement, video, mathematik, abstract, industrial, review, example, department, information, enterprises, informatik, laboratory, introduction, encyclopedia, algorithm, section, available

A.6 Integrating Out Probability Distributions

Here, we show how to integrate out probability distributions using the expectation of a PYP:

$$\begin{aligned} p\big(w_{dn} \,|\, z_{dn} = k, \phi_k\big) &= \int_{\phi'_{dk}} p\big(w_{dn}, \phi'_{dk} \,|\, z_{dn}, \phi_k\big) \\ &= \int_{\phi'_{dk}} p\big(w_{dn} \,|\, z_{dn}, \phi'_{dk}\big)\, p\big(\phi'_{dk} \,|\, \phi_k\big) \\ &= \int_{\phi'_{dk}} \phi'_{dkw_{dn}}\, p\big(\phi'_{dk} \,|\, \phi_k\big) \\ &= \mathbb{E}\Big[\phi'_{dkw_{dn}} \,\Big|\, \phi_k\Big] \\ &= \phi_{kw_{dn}} \qquad\qquad , \end{aligned} \tag{A.17}$$

where $\mathbb{E}[\cdot]$ denotes the expectation value. We note that the last step in Equation A.17 follows from the fact that the expected value of a PYP is the probability vector corresponding to the base distribution of the PYP (when the base distribution is a probability distribution).

Bibliography

Ahmed, A. and Xing, E. P. (2010). Timeline: A dynamic hierarchical Dirichlet process model for recovering birth/death and evolution of topics in text stream. In Grünwald, P. and Spirtes, P., editors, *Proceedings of the Twenty-sixth Conference on Uncertainty in Artificial Intelligence*, UAI 2010, pages 20–29. Corvallis, Oregon, USA. Association for Uncertainty in Artificial Intelligence Press. (cited on page 32)

Aletras, N. and Stevenson, M. (2014). Labelling topics using unsupervised graph-based methods. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL 2014, pages 631–636. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 27)

AlSumait, L., Barbará, D., and Domeniconi, C. (2008). On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking. In Kellenberger, P., editor, *Proceedings of the Eighth IEEE International Conference on Data Mining*, ICDM 2008, pages 3–12. Piscataway, New Jersey, USA. Institute of Electrical and Electronics Engineers. (cited on page 32)

Aula, P. (2010). Social media, reputation risk and ambient publicity management. *Strategy & Leadership*, 38(6):43–49. (cited on page 107)

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, LREC 2010, pages 2200–2204. Paris, France. European Language Resources Association. (cited on page 63)

Baldwin, T., Cook, P., Lui, M., MacKinlay, A., and Wang, L. (2013). How noisy social media text, how diffrnt *[sic]* social media sources? In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, IJCNLP 2013, pages 356–364. Nagoya, Japan. Asian Federation of Natural Language Processing. (cited on pages 107 and 108)

Bernardo, J. M. and Smith, A. F. M. (1994). *Bayesian Theory*. Wiley series in probability and mathematical statistics. John Wiley & Sons, Chichester, England. (cited on page 10)

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*, volume 1. Springer-Verlag, Secaucus, New Jersey, USA. (cited on pages 12 and 15)

Blackwell, D. and MacQueen, J. B. (1973). Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, 1(2):353–355. (cited on page 24)

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4):77–84. (cited on pages 1 and 27)

Blei, D. M., Griffiths, T. L., and Jordan, M. I. (2010). The nested Chinese Restaurant Process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM*, 57(2):7:1–7:30. (cited on pages 37 and 135)

Blei, D. M. and Jordan, M. I. (2003). Modeling annotated data. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR 2003, pages 127–134. New York City, New York, USA. Association for Computing Machinery. (cited on page 113)

Blei, D. M. and Jordan, M. I. (2006). Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143. (cited on page 138)

Blei, D. M. and Lafferty, J. D. (2006). Dynamic topic models. In Cohen, W. W. and Moore, A., editors, *Proceedings of the 23rd International Conference on Machine Learning*, ICML 2006, pages 113–120. New York City, New York, USA. Association for Computing Machinery. (cited on pages 2 and 32)

Blei, D. M. and Lafferty, J. D. (2007). A correlated topic model of science. *The Annals of Applied Statistics*, 1(1):17–35. (cited on page 32)

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022. (cited on pages 1, 27, and 34)

Broersma, M. and Graham, T. (2012). Social media as beat. *Journalism Practice*, 6(3):403–419. (cited on page 107)

Bundschus, M., Yu, S., Tresp, V., Rettinger, A., Dejori, M., and Kriegel, H.-P. (2009). Hierarchical Bayesian models for collaborative tagging systems. In Wang, W., Kargupta, H., Ranka, S., Yu, P. S., and Wu, X., editors, *Proceedings of the Ninth IEEE International Conference on Data Mining*, ICDM 2009, pages 728–733. Piscataway, New Jersey, USA. Institute of Electrical and Electronics Engineers. (cited on pages 32 and 124)

Buntine, W. L. (2002). Variational extensions to EM and multinomial PCA. In Elomaa, T., Mannila, H., and Toivonen, H., editors, *Proceedings of the 13th European Conference on Machine Learning*, ECML 2002, pages 23–34. Berlin, Heidelberg. Springer. (cited on page 28)

Buntine, W. L. and Hutter, M. (2012). A Bayesian view of the Poisson-Dirichlet process. *ArXiv e-prints arXiv:1007.0296v2*. (cited on pages 22, 25, 38, 42, and 51)

Buntine, W. L. and Mishra, S. (2014). Experiments with non-parametric topic models. In Macskassy, S. A., Perlich, C., Leskovec, J., Wang, W., and Ghani, R., editors, *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 2014, pages 881–890. New York City, New York, USA. Association for Computing Machinery. (cited on pages 83, 84, and 95)

Cano Basave, A. E., He, Y., and Xu, R. (2014). Automatic labelling of topic models learned from Twitter by summarisation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL 2014, pages 618–624. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 27)

Carpenter, B. (2004). Phrasal queries with LingPipe and Lucene: Ad hoc genomics text retrieval. In Voorhees, E. M. and Buckland, L. P., editors, *Proceedings of the Thirteenth Text Retrieval Conference*, TREC 2004. Gaithersburg, Maryland, USA. National Institute of Standards and Technology. (cited on page 95)

Chang, J. and Blei, D. M. (2010). Hierarchical relational models for document networks. *The Annals of Applied Statistics*, 4(1):124–150. (cited on pages 29, 82, and 109)

Chen, C., Du, L., and Buntine, W. L. (2011). Sampling table configurations for the hierarchical Poisson-Dirichlet process. In *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part I*, ECML 2011, pages 296–311. Berlin, Heidelberg. Springer-Verlag. (cited on pages 37, 38, 40, and 135)

Çinlar, E. (2011). *Probability and Stochastics*, volume 261. Springer Science & Business Media, New York City, New York, USA. (cited on page 23)

Correa, T., Hinsley, A. W., and de Zúñiga, H. G. (2010). Who interacts on the Web?: The intersection of users' personality and social media use. *Computers in Human Behavior*, 26(2):247–253. (cited on page 107)

Daumé III, H. (2007). HBC: Hierarchical Bayes Compiler. University of Maryland, USA. (cited on page 33)

Davidov, D., Tsur, O., and Rappoport, A. (2010). Enhanced sentiment learning using Twitter hashtags and smileys. In Huang, C. and Jurafsky, D., editors, *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING 2010, pages 241–249. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 56)

De Marneffe, M.-C., MacCartney, B., and Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, LREC 2006, pages 449–454. Paris, France. European Language Resources Association. (cited on page 68)

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38. (cited on pages 12, 16, and 138)

Ding, X., Liu, B., and Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. In Najork, M., Broder, A. Z., and Chakrabarti, S., editors, *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM 2008, pages 231–240. New York City, New York, USA. Association for Computing Machinery. (cited on pages 4 and 56)

Du, L. (2012). *Non-parametric Bayesian methods for structured topic models*. PhD thesis, The Australian National University, Canberra, Australia. (cited on pages 2 and 26)

Du, L., Buntine, W. L., and Jin, H. (2010). A segmented topic model based on the two-parameter Poisson-Dirichlet process. *Machine Learning*, 81(1):5–19. (cited on page 32)

Du, L., Buntine, W. L., and Jin, H. (2012a). Modelling sequential text with an adaptive topic model. In Tsujii, J., Henderson, J., and Pasca, M., editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL 2012, pages 535–545. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on pages 32, 111, and 113)

Du, L., Buntine, W. L., Jin, H., and Chen, C. (2012b). Sequential latent Dirichlet allocation. *Knowledge and Information Systems*, 31(3):475–503. (cited on pages 2 and 32)

Du, L., Buntine, W. L., and Johnson, M. (2013). Topic segmentation with a structured topic model. In Vanderwende, L., Daumé III, H., and Kirchhoff, K., editors, *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT 2013, pages 190–200. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 32)

Eisenstein, J. (2013). What to do about bad language on the internet. In Vanderwende, L., Daumé III, H., and Kirchhoff, K., editors, *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT 2013, pages 359–369. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 108)

Erosheva, E. A. and Fienberg, S. E. (2005). *Bayesian Mixed Membership Models for Soft Clustering and Classification*, pages 11–26. Springer, Berlin, Heidelberg. (cited on page 48)

Fei-Fei, L. and Perona, P. (2005). A Bayesian hierarchical model for learning natural scene categories. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR 2005, pages 524–531. Piscataway, New Jersey, USA. Institute of Electrical and Electronics Engineers. (cited on page 27)

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. Massachusetts Institute of Technology Press, Cambridge, Massachusetts, USA. (cited on page 63)

Fergus, R., Fei-Fei, L., Perona, P., and Zisserman, A. (2005). Learning object categories from Google's image search. In Sebe, N., Lew, M. S., and Huang, T. S., editors, *Proceedings of the Tenth IEEE International Conference on Computer Vision*, volume 2 of *ICCV 2005*, pages 1816–1823. Piscataway, New Jersey, USA. Institute of Electrical and Electronics Engineers. (cited on page 27)

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230. (cited on pages 2 and 24)

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis, Third Edition*. Chapman & Hall/CRC Texts in Statistical Science. CRC Press, Boca Raton, Florida, USA. (cited on pages 10 and 12)

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741. (cited on pages 12 and 14)

Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision. Technical Report CS224N Project Report, Stanford University, California, USA. (cited on pages 56 and 67)

Goldwater, S., Griffiths, T. L., and Johnson, M. (2005). Interpolating between types and tokens by estimating power-law generators. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems 18*, NIPS 2005, pages 459–466. Massachusetts Institute of Technology Press, Cambridge, Massachusetts, USA. (cited on pages 2 and 25)

Goldwater, S., Griffiths, T. L., and Johnson, M. (2011). Producing power-law distributions and damping word frequencies with two-stage language models. *Journal of Machine Learning Research*, 12:2335–2382. (cited on pages 34 and 125)

Green, P. J. and Hastie, D. I. (2009). Reversible jump MCMC. *Genetics*, 155(3):1391–1403. (cited on page 138)

Gruber, A., Weiss, Y., and Rosen-Zvi, M. (2007). Hidden topic Markov models. In Meila, M. and Shen, X., editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, AISTATS 2007, pages 163–170. Brookline, Massachusetts, USA. Microtome Publishing. (cited on page 32)

Haldane, J. B. S. (1932). A note on inverse probability. *Mathematical Proceedings of the Cambridge Philosophical Society*, 28:55–61. (cited on page 19)

Han, B., Cook, P., and Baldwin, T. (2012). Automatically constructing a normalisation dictionary for microblogs. In Tsujii, J., Henderson, J., and Pasca, M., editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL 2012, pages 421–432. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 68)

Han, B., Cook, P., and Baldwin, T. (2013). Lexical normalization for social media text. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(1):5:1–5:27. (cited on page 68)

Han, H., Giles, L., Zha, H., Li, C., and Tsioutsiouliklis, K. (2004). Two supervised learning approaches for name disambiguation in author citations. In Chen, H., Wactlar, H. D., Chen, C., Lim, E., and Christel, M. G., editors, *Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL 2004, pages 296–305. New York City, New York, USA. Association for Computing Machinery. (cited on page 94)

Han, H., Zha, H., and Giles, C. L. (2005). Name disambiguation in author citations using a K-way spectral clustering method. In Marlino, M., Sumner, T., and III, F. M. S., editors, *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL 2005, pages 334–343. New York City, New York, USA. Association for Computing Machinery. (cited on page 94)

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109. (cited on pages 12 and 13)

He, Q., Chen, B., Pei, J., Qiu, B., Mitra, P., and Giles, C. L. (2009). Detecting topic evolution in scientific literature: How can citations help? In Cheung, D. W., Song, I., Chu, W. W., Hu, X., and Lin, J. J., editors, *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM 2009, pages 957–966. New York City, New York, USA. Association for Computing Machinery. (cited on page 32)

He, Y. (2012). Incorporating sentiment prior knowledge for weakly supervised sentiment analysis. *ACM Transactions on Asian Language Information Processing (TALIP)*, 11(2):4:1–4:19. (cited on pages 2, 4, 56, 62, and 63)

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. *ArXiv e-prints 1309.6835v1*. (cited on page 134)

Hjort, N. L., Holmes, C., Müller, P., and Walker, S. G. (2010). *Bayesian Nonparametrics*, volume 28. Cambridge University Press, Cambridge, England. (cited on page 23)

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347. (cited on page 15)

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 1999, pages 50–57. New York City, New York, USA. Association for Computing Machinery. (cited on pages 1 and 28)

Hong, L. and Davison, B. D. (2010). Empirical study of topic modeling in Twitter. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA 2010, pages 80–88. New York City, New York, USA. Association for Computing Machinery. (cited on page 29)

Hospedales, T., Gong, S., and Xiang, T. (2012). Video behaviour mining using a dynamic topic model. *International Journal of Computer Vision*, 98(3):303–323. (cited on page 27)

Hu, M. and Liu, B. (2004). Mining opinion features in customer reviews. In McGuinness, D. L. and Ferguson, G., editors, *Proceedings of the Nineteenth National Conference on Artifial Intelligence*, AAAI 2004, pages 755–760. Palo Alto, California, USA. Association for the Advancement of Artificial Intelligence Press. (cited on page 55)

Ishwaran, H. and James, L. F. (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173. (cited on pages 2 and 24)

Jagarlamudi, J., Daumé III, H., and Udupa, R. (2012). Incorporating lexical priors into topic models. In Bouma, G. and Parmentier, Y., editors, *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL 2012, pages 204–213. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 56)

Jain, S. and Neal, R. M. (2004). A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13(1):158–182. (cited on page 138)

Jaynes, E. T. and Kempthorne, O. (1976). Confidence intervals vs Bayesian intervals. In Harper, W. L. and Hooker, C. A., editors, *Foundations of Probability Theory, Statistical Inference, and Statistical Theories of Science*, pages 175–257. Springer, Dordrecht, Netherlands. (cited on page 11)

Jelinek, F. (1997). *Statistical Methods for Speech Recognition*. Massachusetts Institute of Technology Press, Cambridge, Massachusetts, USA. (cited on page 1)

Jiang, L., Yu, M., Zhou, M., Liu, X., and Zhao, T. (2011). Target-dependent Twitter sentiment classification. In Lin, D., Matsumoto, Y., and Mihalcea, R., editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, ACL-HLT 2011, pages 151–160. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 56)

Jin, O., Liu, N. N., Zhao, K., Yu, Y., and Yang, Q. (2011). Transferring topical knowledge from auxiliary long texts for short text clustering. In Macdonald, C., Ounis, I., and Ruthven, I., editors, *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM 2011, pages 775–784. New York City, New York, USA. Association for Computing Machinery. (cited on page 2)

Jo, Y. and Oh, A. H. (2011). Aspect and sentiment unification model for online review analysis. In King, I., Nejdl, W., and Li, H., editors, *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM 2011, pages 815–824. New York City, New York, USA. Association for Computing Machinery. (cited on page 55)

Johnson, M., Griffiths, T. L., and Goldwater, S. (2007). Adaptor Grammars: A framework for specifying compositional nonparametric Bayesian models. In Schölkopf, B., Platt, J. C., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*, NIPS 2007, pages 641–648. Massachusetts Institute of Technology Press, Cambridge, Massachusetts, USA. (cited on page 33)

Jurafsky, D. and Martin, J. H. (2000). *Speech & Language Processing*. Prentice-Hall, Upper Saddle River, New Jersey, USA. (cited on page 1)

Karimi, S., Yin, J., and Paris, C. (2013). Classifying microblogs for disasters. In Culpepper, S., Zuccon, G., and Sitbon, L., editors, *Proceedings of the 18th Australasian Document Computing Symposium*, ADCS 2013, pages 26–33. New York City, New York, USA. Association for Computing Machinery. (cited on page 107)

Kataria, S., Mitra, P., Caragea, C., and Giles, C. L. (2011). Context sensitive topic models for author influence in document networks. In Walsh, T., editor, *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Three*, IJCAI 2011, pages 2274–2280. Palo Alto, California, USA. Association for the Advancement of Artificial Intelligence Press. (cited on pages 2, 82, and 109)

Kim, D., Kim, S., and Oh, A. (2012). Dirichlet process with mixed random measures: A nonparametric topic model for labeled data. In Langford, J. and Pineau, J., editors, *Proceedings of the 29th International Conference on Machine Learning*, ICML 2012, pages 727–734. New York City, New York, USA. Omnipress. (cited on page 26)

Kinsella, S., Murdock, V., and O'Hare, N. (2011). "I'm eating a sandwich in Glasgow": Modeling locations with tweets. In Cantador, I., Carrero, F. M., Cortizo, J. C., Rosso, P., Schedl, M., and Troyano, J. A., editors, *Proceedings of the 3rd International Workshop on Search and Mining User-generated Contents*, SMUC 2011, pages 61–68. New York City, New York, USA. Association for Computing Machinery. (cited on page 2)

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86. (cited on page 15)

Kwak, H., Lee, C., Park, H., and Moon, S. (2010). What is Twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW 2010, pages 591–600. New York City, New York, USA. Association for Computing Machinery. (cited on page 123)

Landauer, T. K., McNamara, D. S., Dennis, S., and Kintsch, W. (2007). *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum, Mahwah, New Jersey, USA. (cited on page 1)

Lau, J. H., Grieser, K., Newman, D., and Baldwin, T. (2011). Automatic labelling of topic models. In Lin, D., Matsumoto, Y., and Mihalcea, R., editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, ACL-HLT 2011, pages 1536–1545. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on pages 27 and 130)

Li, A. Q., Ahmed, A., Ravi, S., and Smola, A. J. (2014). Reducing the sampling complexity of topic models. In Macskassy, S. A., Perlich, C., Leskovec, J., Wang, W., and Ghani, R., editors, *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 2014, pages 891–900. New York City, New York, USA. Association for Computing Machinery. (cited on page 138)

Li, F., Han, C., Huang, M., Zhu, X., Xia, Y.-J., Zhang, S., and Yu, H. (2010). Structure-aware review mining and summarization. In Huang, C. and Jurafsky, D., editors, *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING 2010. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 56)

Li, T., Zhang, Y., and Sindhwani, V. (2009). A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge. In Su, K., Su, J., and Wiebe, J., editors, *Proceedings of the Joint Conference of the 47th Annual Meeting of* *the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, ACL 2009, pages 244–252. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 56)

Lim, K. W. and Buntine, W. L. (2014a). Bibliographic analysis with the Citation Network Topic Model. In Phung, D. and Li, H., editors, *Proceedings of the Sixth Asian Conference on Machine Learning*, ACML 2014, pages 142–158. Brookline, Massachusetts, USA. Microtome Publishing. (cited on pages 5 and 81)

Lim, K. W. and Buntine, W. L. (2014b). Twitter Opinion Topic Model: Extracting product opinions from tweets by leveraging hashtags and sentiment lexicon. In Li, J., Wang, X. S., Garofalakis, M. N., Soboroff, I., Suel, T., and Wang, M., editors, *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM 2014, pages 1319–1328. New York City, New York, USA. Association for Computing Machinery. (cited on pages 5 and 53)

Lim, K. W. and Buntine, W. L. (2016). Bibliographic analysis on research publications using authors, categorical labels and the citation network. *Machine Learning*, 103(2):185–213. (cited on page 5)

Lim, K. W., Buntine, W. L., Chen, C., and Du, L. (2016). Nonparametric Bayesian topic modelling with the hierarchical Pitman-Yor processes. *International Journal of Approximate Reasoning*, 1(1):1–40. (cited on pages 6 and 33)

Lim, K. W., Chen, C., and Buntine, W. L. (2013). Twitter-Network Topic Model: A full Bayesian treatment for social network and text modeling. In *Advances in Neural Information Processing Systems: Topic Models Workshop*, NIPS Workshop 2013, pages 1–5. Lake Tahoe, Nevada, USA. (cited on pages 6 and 107)

Lim, K. W., Sanner, S., and Guo, S. (2012). On the mathematical relationship between expected n-call@k and the relevance vs. diversity trade-off. In Hersh, W. R., Callan, J., Maarek, Y., and Sanderson, M., editors, *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 2012, pages 1117–1118. New York City, New York, USA. Association for Computing Machinery.

Lin, C. and He, Y. (2009). Joint sentiment/topic model for sentiment analysis. In Cheung, D. W., Song, I., Chu, W. W., Hu, X., and Lin, J. J., editors, *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM 2009, pages 375–384. New York City, New York, USA. Association for Computing Machinery. (cited on pages 32 and 55)

Lindsey, R. V., Headden III, W. P., and Stipicevic, M. J. (2012). A phrase-discovering topic model using hierarchical Pitman-Yor processes. In Tsujii, J., Henderson, J., and Pasca, M., editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL 2012, pages 214–222. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 2)

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167. (cited on pages 1 and 53)

Liu, J. S. (1994). The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89(427):958–966. (cited on page 15)

Liu, L., Tang, J., Han, J., Jiang, M., and Yang, S. (2010). Mining topic-level influence in heterogeneous networks. In Huang, J., Koudas, N., Jones, G. J. F., Wu, X., Collins-Thompson, K., and An, A., editors, *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM 2010, pages 199–208. New York City, New York, USA. Association for Computing Machinery. (cited on page 82)

Liu, S., Li, F., Li, F., Cheng, X., and Shen, H. (2013). Adaptive co-training SVM for sentiment classification on tweets. In He, Q., Iyengar, A., Nejdl, W., Pei, J., and Rastogi, R., editors, *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, CIKM 2013, pages 2079–2088. New York City, New York, USA. Association for Computing Machinery. (cited on page 56)

Liu, Y., Niculescu-Mizil, A., and Gryc, W. (2009). Topic-link LDA: Joint models of topic and author community. In Danyluk, A. P., Bottou, L., and Littman, M. L., editors, *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML 2009, pages 665–672. New York City, New York, USA. Association for Computing Machinery. (cited on pages 29, 82, and 109)

Lloret, E. and Palomar, M. (2012). Text summarisation in progress: A literature review. *Artificial Intelligence Review*, 37(1):1–41. (cited on page 1)

Lloyd, J., Orbanz, P., Ghahramani, Z., and Roy, D. M. (2012). Random function priors for exchangeable arrays with applications to graphs and relational data. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, NIPS 2012, pages 998–1006. Curran Associates, Rostrevor, Northern Ireland. (cited on pages 109 and 113)

Low, A. A. (1991). *Introductory Computer Vision and Image Processing*. McGraw-Hill, New York City, New York, USA. (cited on page 1)

Lu, J., Plataniotis, K. N., and Venetsanopoulos, A. N. (2003). Face recognition using LDA-based algorithms. *IEEE Transactions on Neural Networks*, 14(1):195–200. (cited on page 27)

Lui, M. and Baldwin, T. (2012). langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, ACL 2012, pages 25–30. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on pages 67, 93, and 123)

Lunn, D. J., Thomas, A., Best, N., and Spiegelhalter, D. (2000). WinBUGS - a Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing*, 10(4):325–337. (cited on pages 4 and 33)

Mai, L. C. (2010). Introduction to image processing and computer vision. Technical report, Institute of Information Technology, Hanoi, Vietnam. (cited on page 1)

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York City, New York, USA. (cited on pages 1 and 48)

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. Massachusetts Institute of Technology Press, Cambridge, Massachusetts, USA. (cited on page 1)

Mao, X.-L., Ming, Z.-Y., Zha, Z.-J., Chua, T.-S., Yan, H., and Li, X. (2012). Automatic labeling hierarchical topics. In Chen, X., Lebanon, G., Wang, H., and Zaki, M. J., editors, *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM 2012, pages 2383–2386. New York City, New York, USA. Association for Computing Machinery. (cited on page 27)

Maynard, D., Bontcheva, K., and Rout, D. (2012). Challenges in developing opinion mining tools for social media. In Melero, M., editor, *Proceedings of @NLP can u tag #user_generated_content*, LREC Workshop 2012, pages 15–22. Istanbul, Turkey. (cited on pages 55 and 108)

Mcauliffe, J. D. and Blei, D. M. (2008). Supervised topic models. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T., editors, *Advances in Neural Information Processing Systems 20*, pages 121–128. Curran Associates, Rostrevor, Northern Ireland. (cited on pages 29, 31, and 109)

McCallum, A. K. (2002). MALLET: A machine learning for language toolkit. Cornell University, New York, USA. (cited on pages 69 and 95)

McCord, M. and Chuah, M. (2011). Spam detection on Twitter using traditional classifiers. In *Proceedings of the 8th International Conference on Autonomic and Trusted Computing*, ATC 2011, pages 175–186. Berlin, Heidelberg. Springer-Verlag. (cited on pages 80 and 137)

McCullagh, P. (1984). Generalized linear models. *European Journal of Operational Research*, 16(3):285–292. (cited on page 31)

Mehdad, Y., Carenini, G., Ng, R. T., and Joty, S. R. (2013). Towards topic labeling with phrase entailment and aggregation. In Vanderwende, L., Daumé III, H., and Kirchhoff, K., editors, *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT 2013, pages 179–189. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 130)

Mehrotra, R., Sanner, S., Buntine, W. L., and Xie, L. (2013). Improving LDA topic models for microblogs via tweet pooling and automatic labeling. In Jones, G. J. F., Sheridan, P., Kelly, D., de Rijke, M., and Sakai, T., editors, *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 2013, pages 889–892. New York City, New York, USA. Association for Computing Machinery. (cited on pages 54, 68, 108, 123, 124, 125, 128, and 129)

Mei, Q., Ling, X., Wondra, M., Su, H., and Zhai, C. (2007). Topic sentiment mixture: Modeling facets and opinions in weblogs. In *Proceedings of the 16th International Conference on World Wide Web*, WWW 2007, pages 171–180. New York City, New York, USA. Association for Computing Machinery. (cited on pages 2, 32, and 55)

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092. (cited on pages 12 and 13)

Mimno, D. and McCallum, A. (2007). Mining a digital library for influential authors. In Rasmussen, E. M., Larson, R. R., Toms, E. G., and Sugimoto, S., editors, *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL 2007, pages 105–106. New York City, New York, USA. Association for Computing Machinery. (cited on page 82)

Minka, T. P. (2001). Expectation Propagation for approximate Bayesian inference. In Breese, J. and Koller, D., editors, *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI 2001, pages 362–369. San Francisco, California, USA. Morgan Kaufmann. (cited on pages 16 and 138)

Minka, T. P., Winn, J. M., Guiver, J. P., Webster, S., Zaykov, Y., Yangel, B., Spengler, A., and Bronskill, J. (2014). Infer.NET 2.6. Microsoft Research, Cambridge, UK. (cited on pages 4 and 33)

Moghaddam, S. and Ester, M. (2010). Opinion Digger: An unsupervised opinion miner from unstructured product reviews. In Huang, J., Koudas, N., Jones, G. J. F., Wu, X., Collins-Thompson, K., and An, A., editors, *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM 2010, pages 1825–1828. New York City, New York, USA. Association for Computing Machinery. (cited on page 55)

Moghaddam, S. and Ester, M. (2011). ILDA: Interdependent LDA model for learning latent aspects and their ratings from online product reviews. In Ma, W., Nie, J., Baeza-Yates, R. A., Chua, T., and Croft, W. B., editors, *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 2011, pages 665–674. New York City, New York, USA. Association for Computing Machinery. (cited on pages 4, 55, and 57)

Moghaddam, S. and Ester, M. (2012). On the design of LDA models for aspect-based opinion mining. In Chen, X., Lebanon, G., Wang, H., and Zaki, M. J., editors, *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM 2012, pages 803–812. New York City, New York, USA. Association for Computing Machinery. (cited on pages 54 and 68)

Moon, T. K. (1996). The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60. (cited on page 138)

Murray, I., Adams, R. P., and MacKay, D. J. C. (2010). Elliptical slice sampling. In Teh, Y. W. and Titterington, D. M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, AISTATS 2010, pages 541–548. Brookline, Massachusetts, USA. Microtome Publishing. (cited on pages 116, 120, and 121)

Nakov, P., Kozareva, Z., Ritter, A., Rosenthal, S., Stoyanov, V., and Wilson, T. (2013). SemEval-2013 task 2: Sentiment analysis in Twitter. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics, Volume 2: Seventh International Workshop on Semantic Evaluation*, SemEval 2013, pages 312–320. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 67)

Nallapati, R. M., Ahmed, A., Xing, E. P., and Cohen, W. W. (2008). Joint latent topic models for text and citations. In Li, Y., Liu, B., and Sarawagi, S., editors, *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 2008, pages 542–550. New York City, New York, USA. Association for Computing Machinery. (cited on pages 82 and 109)

Neal, R. M. (2003). Slice sampling. *The Annals of Statistics*, 31(3):705–741. (cited on page 65)

Newman, D., Karimi, S., and Cavedon, L. (2009). External evaluation of topic models. In Kay, J., Thomas, P., and Trotman, A., editors, *Proceedings of the 14th Australasian Document Computing Symposium*, ADCS 2009, pages 11–18. NSW, Australia. University of Sydney. (cited on pages 47 and 128)

Oehlert, G. W. (1992). A note on the delta method. *The American Statistician*, 46(1):27–29. (cited on pages 5 and 89)

Oldham, K. B., Myland, J., and Spanier, J. (2009). *An Atlas of Functions: With Equator, the Atlas Function Calculator*. Springer Science and Business Media, New York City, New York, USA. (cited on page 38)

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. A. (2013). Improved part-of-speech tagging for online conversational text with word clusters. In Vanderwende, L., Daumé III, H., and Kirchhoff, K., editors, *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT 2013, pages 380–390. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 68)

Pak, A. and Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, LREC 2010, pages 1320–1326. Paris, France. European Language Resources Association. (cited on page 56)

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135. (cited on pages 1 and 53)

Pitman, J. (2006). *Combinatorial Stochastic Processes*. Springer-Verlag, Berlin Heidelberg. (cited on page 25)

Pitman, J. and Yor, M. (1997). The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25(2):855–900. (cited on page 24)

Plummer, M. (2003). JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. In Hornik, K., Leisch, F., and Zeileis, A., editors, *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*, DSC 2003. Vienna, Austria. (cited on pages 4 and 33)

Popescu, A.-M. and Etzioni, O. (2005). Extracting product features and opinions from reviews. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT-EMNLP 2005, pages 339–346. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 55)

Qiu, M., Zhu, F., and Jiang, J. (2013). It is not just what we say, but how we say them: LDA-based behavior-topic model. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, SDM 2013, pages 794–802. Philadelphia, Pennsylvania, USA. Society for Industrial and Applied Mathematics. (cited on page 109)

Rabiner, L. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition*. Prentice-Hall, Upper Saddle River, New Jersey, USA. (cited on page 1)

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*, EMNLP 2009, pages 248–256. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 2)

Ritter, A., Clark, S., Mausam, and Etzioni, O. (2011). Named entity recognition in tweets: An experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP 2011, pages 1524–1534. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on pages 54 and 68)

Rosen-Zvi, M., Griffiths, T., Steyvers, M., and Smyth, P. (2004). The author-topic model for authors and documents. In Halpern, J. and Meek, C., editors, *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI 2004, pages 487–494. Arlington, Virginia, USA. Association for Uncertainty in Artificial Intelligence Press. (cited on pages 2, 29, 82, and 108)

Sato, I. and Nakagawa, H. (2010). Topic models with power-law using Pitman-Yor process. In Rao, B., Krishnapuram, B., Tomkins, A., and Yang, Q., editors, *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 2010, pages 673–682. New York City, New York, USA. Association for Computing Machinery. (cited on pages 2 and 34)

Schnober, C. and Gurevych, I. (2015). Combining topic models for corpus exploration: Applying LDA for complex corpus research tasks in a digital humanities project. In *Proceedings of the 2015 Workshop on Topic Models: Post-Processing and Applications*, TM 2015, pages 11–20. New York City, New York, USA. Association for Computing Machinery. (cited on page 138)

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. (2008). Collective classification in network data. *AI Magazine*, 29(3):93–106. (cited on pages 93 and 143)

Sethuraman, J. (1991). A constructive definition of Dirichlet priors. Technical report, Florida State University, USA. (cited on page 24)

Si, X. and Sun, M. (2009). Tag-LDA for scalable real-time tag recommendation. *Journal of Information and Computational Science*, 6(2):1009–1016. (cited on page 31)

Suominen, H., Hanlen, L., and Paris, C. (2014). Twitter for health – building a social media search engine to better understand and curate laypersons' personal experiences. In Neustein, A., editor, *Text Mining of Web-based Medical Content*, chapter 6, pages 133–174. De Gruyter, Berlin, Germany. (cited on page 1)

Suominen, H., Pyysalo, S., Hiissa, M., Ginter, F., Liu, S., Marghescu, D., Pahikkala, T., Back, B., Karsten, H., and Salakoski, T. (2008). Performance evaluation measures for text mining. *Handbook of Research on TextWeb Mining Technologies*, pages 724–747. (cited on page 73)

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307. (cited on pages 4 and 56)

Tang, J., Sun, J., Wang, C., and Yang, Z. (2009). Social influence analysis in large-scale networks. In IV, J. F. E., Fogelman-Soulié, F., Flach, P. A., and Zaki, M. J., editors, *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 2009, pages 807–816. New York City, New York, USA. Association for Computing Machinery. (cited on page 82)

Teh, Y. W. (2006a). A Bayesian interpretation of interpolated Kneser-Ney. Technical Report TRA2/06, National University of Singapore. (cited on pages 43, 65, and 91)

Teh, Y. W. (2006b). A hierarchical Bayesian language model based on Pitman-Yor processes. In Calzolari, N., Cardie, C., and Isabelle, P., editors, *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, COLING-ACL 2006, pages 985–992. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on pages 34 and 60)

Teh, Y. W. and Jordan, M. I. (2010). Hierarchical Bayesian nonparametric models with applications. In Hjort, N. L., Holmes, C., Müller, P., and Walker, S. G., editors, *Bayesian Nonparametrics: Principles and Practice*, chapter 5. Cambridge University Press. (cited on pages 25, 34, 37, and 135)

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. (cited on pages 2 and 29)

Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., and Kappas, A. (2010). Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558. (cited on page 63)

Titov, I. and McDonald, R. T. (2008a). A joint model of text and aspect ratings for sentiment summarization. In McKeown, K., Moore, J. D., Teufel, S., Allan, J., and Furui, S., editors, *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, ACL 2008, pages 308–316. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 55)

Titov, I. and McDonald, R. T. (2008b). Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th International Conference on World Wide Web*, WWW 2008, pages 111–120. New York City, New York, USA. Association for Computing Machinery. (cited on pages 32 and 55)

Tsai, F. S. (2011). A tag-topic model for blog mining. *Expert Systems with Applications*, 38(5):5330–5335. (cited on pages 5, 29, 30, and 109)

Tsur, O., Davidov, D., and Rappoport, A. (2010). ICWSM-A great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In Cohen, W. W. and Gosling, S., editors, *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, ICWSM 2010, pages 162–169. Palo Alto, California, USA. Association for the Advancement of Artificial Intelligence Press. (cited on pages 80 and 137)

Tu, Y., Johri, N., Roth, D., and Hockenmaier, J. (2010). Citation author topic model in expert search. In Huang, C. and Jurafsky, D., editors, *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING 2010, pages 1265–1273. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on pages 2 and 82)

Walck, C. (2007). Handbook on statistical distributions for experimentalists. Technical Report SUF-PFY/96-01, University of Stockholm, Sweden. (cited on pages 17 and 20)

Wallach, H. M. (2006). Topic modeling: Beyond bag-of-words. In Cohen, W. W. and Moore, A., editors, *Proceedings of the 23rd International Conference on Machine Learning*, ICML 2006, pages 977–984. New York City, New York, USA. Association for Computing Machinery. (cited on page 32)

Wallach, H. M., Mimno, D. M., and McCallum, A. (2009a). 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*, NIPS 2009, pages 1973–1981. Curran Associates, Rostrevor, Northern Ireland. (cited on page 42)

Wallach, H. M., Murray, I., Salakhutdinov, R., and Mimno, D. (2009b). Evaluation methods for topic models. In Danyluk, A. P., Bottou, L., and Littman, M. L., editors, *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML 2009, pages 1105–1112. New York City, New York, USA. Association for Computing Machinery. (cited on pages 46 and 125)

Wang, C., Blei, D., and Heckerman, D. (2008). Continuous time dynamic topic models. In McAllester, D. and Myllymaki, P., editors, *Proceedings of the Twenty-Fourth*

Conference Conference on Uncertainty in Artificial Intelligence, UAI 2008, pages 579–586. Corvallis, Oregon, USA. Association for Uncertainty in Artificial Intelligence Press. (cited on page 32)

Wang, C. and Blei, D. M. (2012). A split-merge MCMC algorithm for the hierarchical Dirichlet process. *ArXiv e-prints 1201.1657v1*. (cited on page 138)

Wang, H., Zhang, D., and Zhai, C. (2011a). Structural topic model for latent topical structure analysis. In Lin, D., Matsumoto, Y., and Mihalcea, R., editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, ACL-HLT 2011, pages 1526–1535. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 32)

Wang, X. and McCallum, A. (2006). Topics over time: A non-Markov continuous-time model of topical trends. In Grossman, R., Bayardo, R. J., and Bennett, K. P., editors, *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 2006, pages 424–433. New York City, New York, USA. Association for Computing Machinery. (cited on page 32)

Wang, X., McCallum, A., and Wei, X. (2007). Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In Kawada, S., editor, *Proceedings of the Seventh IEEE International Conference on Data Mining*, ICDM 2007, pages 697–702. Piscataway, New Jersey, USA. Institute of Electrical and Electronics Engineers. (cited on page 32)

Wang, X., Wei, F., Liu, X., Zhou, M., and Zhang, M. (2011b). Topic sentiment analysis in Twitter: A graph-based hashtag sentiment classification approach. In Macdonald, C., Ounis, I., and Ruthven, I., editors, *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM 2011, pages 1031–1040. New York City, New York, USA. Association for Computing Machinery. (cited on page 2)

Wei, X. and Croft, W. B. (2006). LDA-based document models for ad-hoc retrieval. In Efthimiadis, E. N., Dumais, S. T., Hawking, D., and Järvelin, K., editors, *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 2006, pages 178–185. New York City, New York, USA. Association for Computing Machinery. (cited on page 2)

Weng, J., Lim, E.-P., Jiang, J., and He, Q. (2010). TwitterRank: Finding topic-sensitive influential Twitterers. In Davison, B. D., Suel, T., Craswell, N., and Liu, B., editors, *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM 2010, pages 261–270. New York City, New York, USA. Association for Computing Machinery. (cited on pages 29 and 82)

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT-EMNLP 2005, pages 347–354. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 63)

Wood, F. and Teh, Y. W. (2009). A hierarchical nonparametric Bayesian approach to statistical language model domain adaptation. In van Dyk, D. A. and Welling, M., editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, AISTATS 2009, pages 607–614. Brookline, Massachusetts, USA. Microtome Publishing. (cited on page 26)

Yan, X., Guo, J., Lan, Y., and Cheng, X. (2013). A biterm topic model for short texts. In Schwabe, D., Almeida, V. A. F., Glaser, H., Baeza-Yates, R. A., and Moon, S. B., editors, *Proceedings of the 22nd International Conference on World Wide Web*, WWW 2013, pages 1445–1456. Geneva, Switzerland. International World Wide Web Conferences Steering Committee. (cited on pages 55 and 109)

Yang, J. and Leskovec, J. (2011). Patterns of temporal variation in online media. In King, I., Nejdl, W., and Li, H., editors, *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM 2011, pages 177–186. New York City, New York, USA. Association for Computing Machinery. (cited on pages 67 and 123)

Zhao, W. X. and Jiang, J. (2011). An empirical comparison of topics in Twitter and traditional media. Technical report, Singapore Management University, Singapore. (cited on page 29)

Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E.-P., Yan, H., and Li, X. (2011). Comparing Twitter and traditional media using topic models. In Clough, P. D., Foley, C., Gurrin, C., Jones, G. J. F., Kraaij, W., Lee, H., and Murdock, V., editors, *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, ECIR 2011, pages 338–349. Berlin, Heidelberg. Springer-Verlag. (cited on pages 29, 54, 107, and 109)

Zhao, W. X., Jiang, J., Yan, H., and Li, X. (2010). Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2010, pages 56–65. Stroudsburg, Pennsylvania, USA. Association for Computational Linguistics. (cited on page 55)

Zheng, B., McLean, D. C., and Lu, X. (2006). Identifying biological concepts from a protein-related corpus with a probabilistic topic model. *BMC bioinformatics*, 7(1):1–10. (cited on page 27)

Zhu, X., Blei, D., and Lafferty, J. (2006). TagLDA: Bringing document structure knowledge into topic models. Technical Report TR-1553, University of Wisconsin, USA. (cited on page 31)

Zhu, Y., Yan, X., Getoor, L., and Moore, C. (2013). Scalable text and link analysis with mixed-topic link models. In Dhillon, I. S., Koren, Y., Ghani, R., Senator, T. E., Bradley, P., Parekh, R., He, J., Grossman, R. L., and Uthurusamy, R., editors, *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 2013, pages 473–481. New York City, New York, USA. Association for Computing Machinery. (cited on pages 82, 85, 95, 99, 105, and 109)