JMLR: Workshop and Conference Proceedings 39:142–158, 2014 ACML 2014

Bibliographic Analysis with the Citation Network Topic Model

Kar Wai Lim KARWAI.LIM@ANU.EDU.AU
Australian National University, Canberra, Australia
NICTA, Canberra, Australia

Wray Buntine WRAY.BUNTINE@MONASH.EDU
Monash University, Clayton, Australia

Editor: Dinh Phung and Hang Li

Abstract

Bibliographic analysis considers author's research areas, the citation network and paper content among other things. In this paper, we combine these three in a topic model that produces a bibliographic model of authors, topics and documents using a non-parametric extension of a combination of the Poisson mixed-topic link model and the author-topic model. We propose a novel and efficient inference algorithm for the model 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.

Keywords: author-citation network, topic model, Bayesian non-parametric

1. Introduction

Models of bibliographic data need to consider many kinds of information. Articles are usually accompanied by metadata, for example, authors, publication data, 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 for topic models, and the problem is confounded when also modelling author-topic relationships.

In this paper, we propose a topic model to jointly model authors' topic preferences, text content and the citation network. The model is a non-parametric extension of previous models discussed in Section 2. We derive a novel algorithm that allows the probability vectors in the model to be integrated out, using simple assumptions and approximations, which give Markov chain Monte Carlo (MCMC) inference via discrete sampling. Section 3, 4 and 5 detail our model and its inference algorithm. Applying our model on research publication data, we demonstrate the model's improved performance, on both model fitting and a clustering task, compared to baselines. We describe the datasets used in Section 6 and report on experiments in Section 7. Additionally, we qualitatively analyse the inference results produced by our model. We find that the topics returned have high comprehensibility.

2. Related Work

Latent Dirichlet Allocation (LDA) is the simplest Bayesian topic model used in modelling text, which also allows easy learning of the model. Teh and Jordan (2010) proposed the *Hierarchical Dirichlet process* (HDP) LDA, which utilises the Dirichlet process (DP) as a non-parametric 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 (Sato and Nakagawa, 2010).

Variants of LDA allow incorporating more aspects of a particular task and 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 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 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 (Lim et al., 2013) models the author ("follower") network based on author topic vectors, but using a Gaussian process to model the network. Note that our work considers the author-document-citation of Liu et al. (2009) using the techniques developed in Lim et al. (2013), but using 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.

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 discuss the topic model part of CNTM where the citations are not considered, which will be used for comparison later in Section 7. The full graphical model for CNTM is displayed in Figure 1. To clarify the notations used in this paper, *variables that are without subscript represent a collection of variables of the same notation.* For example, w_d would represent all the words in document d, that is, $w_d = \{w_{d1}, \ldots, w_{dN_d}\}$ where N_d is the number of words in document d; and w represents all words in a corpus, $w = \{w_1, \ldots, w_D\}$, where D is the number of documents.

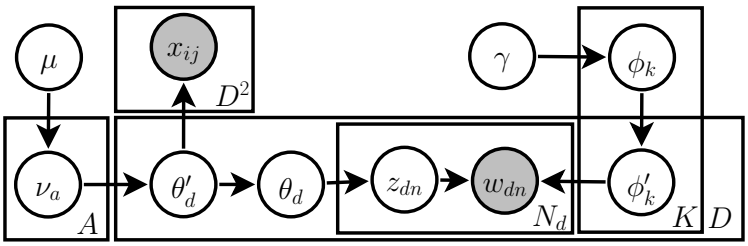

Figure 1: Graphical model for CNTM. The box on the top left with D^2 entries is the citation network on documents represented as a Boolean matrix. The remainder is a non-parametric author-topic model where the A authors on the left have topic vectors that influence the D document topic vectors. The K topics, shown in the top right, have bursty modelling following Buntine and Mishra (2014).

3.1. Hierarchical Pitman-Yor Topic Model

The CNTM uses the *Griffiths-Engen-McCloskey* (GEM) (Pitman, 1996) distribution to generate probability vectors and the *Pitman-Yor process* (PYP) (Teh, 2006b) to generate probability vectors given another probability vector (called *mean* or *base* distribution). Both GEM and PYP are parameterised by a discount parameter α and a concentration parameter β. PYP is additionally parameterised by a base distribution H, which is also the mean of the PYP. Note that the GEM distribution is equivalent to a PYP with a base distribution that generates an ordered integer label.

In modelling authors, CNTM modifies the approach of the author-topic model (Rosen-Zvi et al., 2004), 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 written more by the first author, excepting when the order is alphabetical. An approximation we make in this work is that the first author is dominant. We could model the influence of each author on a publication, say, using a Dirichlet distribution, but we found that considering only the first author gives a simpler learning algorithm and cleaner results.

IN CNTM, 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:

$$\mu \sim \text{GEM}(\alpha^{\mu}, \beta^{\mu}) \ , \qquad \nu_a|\mu \sim \text{PYP}(\alpha^{\nu_a}, \beta^{\nu_a}, \mu) \quad .$$

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:

$$\theta'_d|a_d, \nu \sim \text{PYP}(\alpha^{\theta'_d}, \beta^{\theta'_d}, \nu_{a_d}) \ , \qquad \theta_d|\theta'_d \sim \text{PYP}(\alpha^{\theta_d}, \beta^{\theta_d}, \theta'_d) \quad .$$

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 to counter the relatively low amount of citations compared to the text. More details on the motivation of such modelling is presented in the supplementary materials.

For the vocabulary side, we generate a background word distribution γ, where H^γ is a discrete uniform vector of length $|\mathcal{V}|$ and $\mathcal{V}$ is the set of distinct word tokens observed. Then, we sample a topic-word distribution ϕ_k for each topic k, with γ as the base distribution:

$$\gamma \sim \text{PYP}(\alpha^\gamma, \beta^\gamma, H^\gamma) \ , \qquad \phi_k|\gamma \sim \text{PYP}(\alpha^{\phi_k}, \beta^{\phi_k}, \gamma) \quad .$$

Modelling word burstiness (Buntine and Mishra, 2014) is important since, as shown in Section 6, words in a document are likely to repeat in the document. This is addressed by making topics bursty, so each document only focuses on a subset of words in the topic. To generate ϕ'_{dk} for each topic k in document d:

$$\phi'_{dk}|\phi_k \sim \text{PYP}(\alpha^{\phi'_{dk}}, \beta^{\phi'_{dk}}, \phi_k) \quad .$$

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}.

$$z_{dn}|\theta_d \sim \text{Discrete}(\theta_d) \ , \qquad w_{dn}|z_{dn}, \phi'_d \sim \text{Discrete}(\phi'_{dz_{dn}}) \quad .$$

Note that w includes words from title and abstract, but not the full article of a publication. This is because title and abstract provide a good summary of a publication's topics, while the full article contains too much detail.

3.2. Citation Network Poisson Model

In CNTM, we assume that the citations are generated based on the topics relevant to the publications' using the degree-corrected version of the 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}:

$$x_{ij}|\lambda_{ij} \sim \text{Poisson}(\lambda_{ij}) \ , \qquad \lambda_{ij} = \lambda_i^+ \lambda_j^- \textstyle\sum_k \lambda_k^T \theta'_{ik} \theta'_{jk} \quad . \tag{1}$$

Here, λ_i^+ is the propensity of document i to cite and λ_j^- represents the popularity of cited document j and λ_k^T scales the k-th topic. Hence, a citation from document i to document j is more likely when these documents are having relevant topics. The Poisson distribution[1] is used instead of a Bernoulli because it leads to dramatically reduced complexity in analysis.

4. Model Representation and Posterior

Before presenting the posterior used to develop the MCMC sampler, we briefly review handling of hierarchical PYP models in Section 4.1. We cannot provide an adequately detailed review in this paper, thus we present the main ideas.

1. Note that Poisson distribution is similar to the Bernoulli distribution when the mean parameter is small.

4.1. Modelling with Hierarchical PYPs

The key to efficient Gibbs sampling with PYPs is to marginalise out the probability vectors (*e.g.* topic distributions) in the model and record various associated counts instead, thus yielding a collapsed sampler. While a common approach here is to use the hierarchical Chinese Restaurant Process (CRP) of Teh and Jordan (2010), we use another representation that requires no dynamic memory and has better inference efficiency (Chen et al., 2011).

We denote $f(\mathcal{N})$ as the marginalised likelihood associated with the probability vector $\mathcal{N}$. Since the vector is marginalised out, the marginalised likelihood is in terms of — using the CRP terminology — the *customer counts* $c^{\mathcal{N}} = (\cdots, c_k^{\mathcal{N}}, \cdots)$ and the *table counts* $t^{\mathcal{N}} = (\cdots, t_k^{\mathcal{N}}, \cdots)$. The customer count $c_k^{\mathcal{N}}$ corresponds to the number of data points (*e.g.* words) assigned to group k (*e.g.* topic) for variable $\mathcal{N}$. Here, the *table counts* $t^{\mathcal{N}}$ represent the subset of $c^{\mathcal{N}}$ that gets passed up the hierarchy (as customers for the parent probability vector of $\mathcal{N}$). We also denote $C^{\mathcal{N}} = \sum_k c_k^{\mathcal{N}}$ as the total customer counts for node $\mathcal{N}$, and similarly, $T^{\mathcal{N}} = \sum_k t_k^{\mathcal{N}}$ is the total table counts. The marginalised likelihood is:

$$f(\mathcal{N}) = \frac{(\beta^{\mathcal{N}}|\alpha^{\mathcal{N}})_{T^{\mathcal{N}}}}{(\beta^{\mathcal{N}})_{C^{\mathcal{N}}}} \prod_k S^{c_k^{\mathcal{N}}}_{t_k^{\mathcal{N}},\alpha^{\mathcal{N}}} \quad , \qquad \text{for } \mathcal{N} \sim \text{PYP}(\alpha^{\mathcal{N}}, \beta^{\mathcal{N}}, \mathcal{P}) \quad . \tag{2}$$

$S^x_{y,\alpha}$ is the generalised Stirling number; both $(x)_C$ and $(x|y)_C$ denote the Pochhammer symbol (rising factorial), see Buntine and Hutter (2012) for details. Note the GEM distribution behaves like a PYP in which the table count $t_k^{\mathcal{N}}$ is always 1 for non-zero $c_k^{\mathcal{N}}$.

The innovation of Chen et al. (2011) was to notice that sampling with Equation 2 directly led to poor performance due to inadequate mixing. They introduce a new Bernoulli *indicator variable* $u_k^{\mathcal{N}}$ for each customer who has contributed a "+1" to $c_k^{\mathcal{N}}$. A value $u_k^{\mathcal{N}} = 1$ indicates that the customer has opened a new table, which also means the customer has also contributed a "+1" to $t_k^{\mathcal{N}}$ and thus has been passed up the hierarchy to the parent variable $\mathcal{P}$. The process repeats at the parent node because the "+1" to $t_k^{\mathcal{N}}$ is inherited as a "+1" to $c_k^{\mathcal{P}}$, and thus we now need to consider the value of $u_k^{\mathcal{P}}$. If $u_k^{\mathcal{N}} = 0$ then a "+1" was not inherited and a corresponding $u_k^{\mathcal{P}}$ does not exist. The use of indicator variables has been empirically shown to lead to better mixing of the samplers.

Note that even though the probability vectors are integrated out and not explicitly stored, they can easily be estimated from the associated counts. The probability vector $\mathcal{N}$ is estimated from the counts and parent probability vector $\mathcal{P}$ using standard CRP estimation:

$$\mathcal{N} = \left(\cdots, \frac{(\alpha^{\mathcal{N}} T^{\mathcal{N}} + \beta^{\mathcal{N}})\mathcal{P}_k + c_k^{\mathcal{N}} - \alpha^{\mathcal{N}} T_k^{\mathcal{N}}}{\beta^{\mathcal{N}} + C^{\mathcal{N}}}, \cdots \right) \quad . \tag{3}$$

4.2. Likelihood for the Hierarchical PYP Topic Model

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. Variables $\mathbf{W}, \mathbf{T}$ and $\mathbf{C}$ are similarly defined, that is, they denote the set of all words, table counts and customer counts respectively. Additionally, we denote ζ as the set of all hyperparameters (such as the α's). With the probability vectors replaced by the counts, the likelihood of the topic

model can be written — in terms of $f(\cdot)$ — as $p(\mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}|\zeta) \propto$

$$f(\mu)\left(\prod_{a=1}^{A} f(\nu_a)\right)\left(\prod_{d=1}^{D} f(\theta'_d)\, f(\theta_d) \prod_{k=1}^{K} f(\phi'_{dk})\right)\left(\prod_{k=1}^{K} f(\phi_k)\right) f(\gamma)\left(\prod_{v}\left(\frac{1}{|\mathcal{V}|}\right)^{t_v^{\gamma}}\right) \quad . \quad (4)$$

Note that the last term in Equation 4 corresponds to the parent probability vector of γ (see Section 3.1), and v indexes the unique word tokens in vocabulary set $\mathcal{V}$.

4.3. Likelihood for the Citation Network Poisson Model

For the citation network, the Poisson likelihood for each x_{ij} uses the definition of λ_{ij} in Equation 1. Note that the term $x_{ij}!$ is dropped due to the limitation of the data that $x_{ij} \in \{0, 1\}$, thus $x_{ij}!$ is evaluated to 1. With conditional independence of x_{ij}, the joint likelihood for the whole citation network $\mathbf{X} = \{x_{11}, \cdots, x_{DD}\}$ can be written as

$$p(\mathbf{X}|\lambda, \theta') = \left(\prod_i (\lambda_i^+)^{g_i^+} (\lambda_i^-)^{g_i^-}\right) \prod_{ij}\left(\sum_k \lambda_k^T \theta'_{ik} \theta'_{jk}\right)^{x_{ij}} \exp\left(-\sum_{ijk} \lambda_i^+ \lambda_j^- \lambda_k^T \theta'_{ik} \theta'_{jk}\right) \quad ,$$

where g_i^+ is the number of citations for publication i, $g_i^+ = \sum_j x_{ij}$, and g_i^- is the number of times publication i being cited, $g_i^- = \sum_j x_{ji}$. We also make a simplifying assumption[2] that $x_{ii} = 1$ for all documents i, that is, all publications are treated as self-cited.

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.

5. Inference Techniques

Here, we derive the Markov chain Monte Carlo (MCMC) algorithms for learning the Citation Network Topic Model. We first detail the Gibbs sampler for the topic model and then discuss the Metropolis-Hastings (MH) algorithm for the citation network. The full inference procedure is performed by alternating between the Gibbs sampler and the MH algorithm.

5.1. Collapsed Gibbs Sampler for the Hierarchical PYP Topic Model

To jointly sample the words' topic and the associated counts in the CNTM, we use a collapsed Gibbs sampler designed for the PYP (Chen et al., 2011). The concept of the sampler is analogous to LDA, which consists of decrementing the counts associated with a word, sampling the respective new topic assignment for the word, and incrementing the associated counts. Our collapsed Gibbs sampler is more complicated than LDA. In particular, we have to consider the indicators $u_k^{\mathcal{N}}$ described in Section 4.1 operating on the hierarchy of PYPs.

The sampler proceeds by considering the latent variables associated with a given word w_{dn}. First, we decrement out the effects of the latent variables, the topic $z_{dn} = k$ and the chain of indicator variables $u_k^{\theta_d}, u_k^{\theta'_d}, u_k^{\nu_{a_d}}, u_k^{\mu}$ (where they exist). After decrementing,

2. Technically, defining x_{ii} allows us to rewrite the joint likelihood into another form for efficient caching.

we jointly sample a new topic z_{dn} and the associated indicators (which contribute "+1" to counts) for word w_{dn} from their joint conditional posterior distribution:

$$p(z_{dn}, \mathbf{T}, \mathbf{C}|\mathbf{Z}^{-dn}, \mathbf{W}, \mathbf{T}^{-dn}, \mathbf{C}^{-dn}, \zeta) = \frac{p(\mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}|\zeta)}{p(\mathbf{Z}^{-dn}, \mathbf{W}, \mathbf{T}^{-dn}, \mathbf{C}^{-dn}|\zeta)} \quad . \tag{5}$$

where the superscript $\square^{-dn}$ indicates that the topic z_{dn}, indicators and the associated counts for word w_{dn} are not observed in the respective sets, *i.e.* the state after decrement. The modularised likelihood of Equation 4 allows the conditional posterior (Equation 5) to be computed easily, since it simplifies to ratios of likelihood $f(\cdot)$, which simplifies further as the counts differ by at most 1 during sampling. For instance, the ratio of the Pochhammer symbols, $(x|y)_{C+1}/(x|y)_C$, simplifies to $x + Cy$, 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).

Sampling a new $z_{dn} = k$ corresponds to incrementing the counts for variable θ_d, that is, "+1" to $c^{\theta_d}_k$ and *possibly* also "+1" to $t^{\theta_d}_k$. If $t^{\theta_d}_k$ is incremented, then $c^{\theta'_d}_k$ will be incremented too but $t^{\theta'_d}_k$ may or may not be, as dictated by the sampled indicators u_k. The process is repeated until the root μ, since μ is GEM distributed, incrementing t^{μ}_k is equivalent to sampling a *new* topic, *i.e.* the number of topics increase by 1. Procedure on the vocabulary side (ϕ *etc.*) is similar.

5.2. Metropolis-Hastings Algorithm for Citation Network

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 and GEM 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 1, a citing topic y_{ij} is jointly Poisson with x_{ij}:

$$x_{ij}, y_{ij} = k|\lambda, \theta' \sim \text{Poisson}\left(\lambda^+_i \lambda^-_j \lambda^T_k \theta'_{ik} \theta'_{jk}\right) \quad . \tag{6}$$

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

$$p(\mathbf{X}, \mathbf{Y}|\lambda, \theta') \propto \prod_i (\lambda^+_i)^{g^+_i} (\lambda^-_i)^{g^-_i} \prod_k \left(\lambda^T_k\right)^{\frac{1}{2}\sum_i h_{ik}} \prod_{ik} {\theta'_{ik}}^{h_{ik}} \exp\left(- \sum_{ij} \lambda^+_i \lambda^-_j \lambda^T_{y_{ij}} \theta'_{iy_{ij}} \theta'_{jy_{ij}} \right)$$

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.

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 $p(\mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}|\zeta)$ where they correspond to additional counts for $c^{\theta'_i}$, with h_{ik} added to $c^{\theta'_i}_k$. 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 ($\mathbf{C}$ plus network counts) as $\mathbf{C}^+$. For the exponential term, we use Delta method approximation, $\int f(\theta) \exp(-g(\theta))\, \mathrm{d}\theta \approx \exp(-g(\hat{\theta})) \int f(\theta)\, \mathrm{d}\theta$, 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 supplementary material). The approximate full posterior of CNTM can then be written as $p(\mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}, \mathbf{X}, \mathbf{Y}|\lambda, \zeta) \approx$

$$p(\mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}^+|\zeta) \prod_i (\lambda_i^+)^{g_i^+} (\lambda_i^-)^{g_i^-} \prod_k \left(\lambda_k^T\right)^{\frac{1}{2}\sum_i h_{ik}} \exp\left(- \sum_{ij} \lambda_i^+ \lambda_j^- \lambda_{y_{ij}}^T \hat{\theta}'_{iy_{ij}} \hat{\theta}'_{jy_{ij}} \right) \quad (7)$$

The MH algorithm can be summarised in three steps: estimate the document topic prior θ', propose a new citing topic y_{ij} from Equation 6, and accept or reject the proposed y_{ij} following an MH scheme with Equation 7. We present the details of the MH sampler in the supplementary material. 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 a documents' topic.

5.3. Hyperparameter Sampling

Hyperparameter sampling for the priors are important (Wallach et al., 2009). In our inference algorithm, we sample the concentration parameters β of all PYPs with an auxiliary variable sampler (Teh, 2006a)[3], but leaving the discount parameters α fixed. We do not sample the α due to the coupling of the parameter with the Stirling numbers cache.

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. 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.

$$\begin{aligned}
(\lambda_i^+|\mathbf{X}, \lambda^-, \lambda^T\theta') &\sim \text{Gamma}\left(\epsilon_0 + g_i^+, \epsilon_1 + \textstyle\sum_k \lambda_k^T \theta'_{ik} \sum_j \lambda_j^- \theta'_{jk}\right) \quad , \\
(\lambda_i^-|\mathbf{X}, \lambda^+, \lambda^T\theta') &\sim \text{Gamma}\left(\epsilon_0 + g_i^-, \epsilon_1 + \textstyle\sum_k \lambda_k^T \theta'_{ik} \sum_j \lambda_j^+ \theta'_{jk}\right) \quad , \\
(\lambda_k^T|\mathbf{X}, \mathbf{Y}, \lambda^+, \lambda^-, \theta') &\sim \text{Gamma}\left(\epsilon_0 + \tfrac{1}{2}\textstyle\sum_i h_{ik}, \epsilon_1 + \lambda_k^T (\sum_j \lambda_j^+ \theta'_{jk})(\sum_j \lambda_j^- \theta'_{jk})\right) \quad .
\end{aligned}$$

In this paper, we apply vague priors to the hyperpriors by setting $\epsilon_0 = \epsilon_1 = 1$.

We summarise the full inference algorithm for the CNTM in Algorithm 1.

6. Data

We perform our experiments on subsets of CiteSeer$^{\text{X}}$ data[4] which consists of scientific publications. Each publication from CiteSeer$^{\text{X}}$ is accompanied by *title*, *abstract*, *keywords*, *authors*, *citations* and other metadata. We prepare three publication datasets from CiteSeer$^{\text{X}}$ for evaluations. The first dataset corresponds to Machine Learning (ML) publications,

3. We outline the hyperparameter sampling for concentration parameters in the supplementary material.
4. http://citeseerx.ist.psu.edu/

Algorithm 1 Inference Algorithm for the Citation Network Topic Model

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:
   i. Decrement the counts associated with z_{dn} and w_{dn} .
   ii. Blocked sample a new topic z_{dn} and associated $\mathbf{T}$ and $\mathbf{C}$ from Equation 5.
3. For each citation x_{ij}:
   i. Decrement the network counts associated with x_{ij} and y_{ij} .
   ii. Sample a new citing topic y_{ij} from the joint posterior of Equation 6.
   iii. Accept or reject the sampled y_{ij} with an MH scheme using Equation 7.
4. Update the hyperparameters β, λ^+, λ^- and λ^T.
5. Repeat steps 2-4 until the model converges or a fix number of iterations reached.

which are queried from CiteSeer$^{\text{X}}$ using the keywords from Microsoft Academic Search[5]. The ML dataset contains 139,227 publications.

Our second dataset corresponds to publications from 10 distinct research areas: *agriculture*, *archaeology*, *biology*, *computer science*, *financial economics*, *industrial engineering*, *material science*, *petroleum chemistry*, *physics* and *social science*. The query words for these 10 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 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 paper.

The keywords used to create the datasets are obtained from Microsoft Academic Search, and are listed in the supplementary material. For the clustering evaluation in Section 7.1.2, 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 publication's 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 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 make use of existing datasets from LINQS (Sen et al., 2008)[6] to facilitate comparison with existing work. In particular, we use their CiteSeer, Cora and PubMed datasets. Their CiteSeer data consists of Artificial Intelligence publications and hence we name the dataset AI in this paper. Although these datasets are small, they are fully labelled and thus useful for clustering evaluation. However,

5. http://academic.research.microsoft.com/
6. http://linqs.cs.umd.edu/projects/projects/lbc/

they do not come with additional metadata such as the authors. Note that the AI and Cora datasets are presented as Boolean matrices, *i.e.* the word counts information is lost and all words in a document are assumed to occur only once. Although this representation is less useful for topic modelling, we still use them for the sake of comparison. Also note that the word counts were converted to TF-IDF in the PubMed dataset, so we recover the word counts using a reasonable assumption, see supplementary material for the recovery process. In Table 1, we present a summary of the datasets used in this paper.

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

Table 1: Summary of the datasets used in the paper, showing the number of publications, citations, authors, unique word tokens, the average number of words in each document, and the last column is the average percentage of unique words repeated in a document. Note: author information is not available on the last three datasets.

6.1. Data Noise Removal

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.

Moreover, the author information is not consistently presented in CiteSeer^{X}. 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 paper. Occasionally, institutions are mistakenly treated as authors in CiteSeer^{X} data, example includes *American Mathematical Society* and *Technische Universität München*. In this case, we simply remove the incorrect authors using a list of exclusion words[7] for authors.

6.2. Text Preprocessing

Here, we discuss the preprocessing pipeline adopted for the *queried* datasets (LINQS data were already processed). First, since publication text contains many technical terms that

7. The list of exclusion words is presented in the supplementary material.

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`[8]. In this paper, 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. More specifically, we use the stop words list from `MALLET`[9], 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 threshold 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. Experiments

In this section, we describe experiments that compare the CNTM against several baseline topic models. The baselines are HDP-LDA with burstiness (Buntine and Mishra, 2014), a non-parametric extension of the ATM, the Poisson mixed-topic link model (PMTLM) (Zhu et al., 2013) and the CNTM without the citation network. We evaluate these models quantitatively with goodness-of-fit and clustering measures. We qualitatively analyse the topics produced and perform topic analysis on the authors. Additionally, we experiment on merging authors who have low number of publications and grouping them based on categories. This gives us a semi-supervised topic modelling in which some labels are known for authors who do not publish much. Finally, we present a discussion on the algorithm running time and convergence analysis in the supplementary material.

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 fix the α to 0.01 for all other PYPs. Note that the number of topics grow with data in non-parametric 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 particular, we set the number of topics cap to 20 for the ML dataset, 50 for M10 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 AI, Cora and PubMed datasets, we *fix* the number of topics to 6, 7 and 3 respectively simply for comparison against 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 1,000 iterations, this is so the topics can be learned first. This gives a faster learning algorithm and also allows us to assess the "*value-added*" by the citation network to topic modelling[10]. We repeat each experiment five times to reduce the estimation error of the evaluation measures.

8. http://alias-i.com/lingpipe/
9. http://mallet.cs.umass.edu/
10. This is elaborated further in the supplementary material with likelihood comparison.

7.1. Quantitative Results

7.1.1. Goodness-of-fit and Perplexity

Perplexity is a popular metric used to evaluate the goodness-of-fit of a topic model. Perplexity is negatively related to the likelihood of the observed words given the model, and lower is better. Perplexity, estimated using document completion, is given as:

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

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

$$p(w_{dn}|\theta_d,\phi) = \sum_k p(w_{dn}|z_{dn}=k,\phi_k)\,p(z_{dn}=k|\theta_d) = \sum_k \phi_{kw_{dn}}\theta_{dk} \quad .$$

The topic distribution θ is unknown for the test documents. Instead of using half of the text in the test set to estimate θ, which is a standard practice, we used only the words from the title to estimate θ. One of the reasons behind this is that although title is usually short, it is a good indicator of topic. Moreover, using only the title allows more words to be used to calculate the perplexity. The technical details on estimating θ is presented in the supplementary material. Note that the perplexity estimate is unbiased since the data used in estimating θ is not used for evaluation.

We present the perplexity result in Table 2, which clearly shows the significantly[11] better performance of CNTM against the baselines. Inclusion of citation information also provides significant improvement for model fitting, as shown in the comparison of CNTM with and without network component.

	ML		M10	
	Train	Test	Train	Test
Bursty HDP-LDA	4904.24 ± 71.34	4992.94 ± 65.57	1959.36 ± 32.77	2265.18 ± 68.19
Non-parametric ATM	2238.19 ± 12.22	2460.28 ± 11.34	1562.85 ± 18.11	1814.03 ± 23.18
CNTM w/o network	1918.21 ± 4.31	2057.61 ± 3.56	912.69 ± 10.94	1186.11 ± 8.32
CNTM w network	**1851.82** ± 8.50	**1990.78** ± 11.36	**824.04** ± 11.96	**1048.33** ± 21.39

Table 2: Perplexity for the train and test documents on ML and M10, lower is better.

7.1.2. Document Clustering

Next, we evaluate the clustering ability of the topic models. Recall that topic models assign a topic to each word in a document, essentially performing a soft clustering in which the membership is given by the document-topic distribution θ. For the following evaluation, 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 corresponds to the topic that has the highest probability in a topic distribution $\mathcal{N}$.

11. In this paper, significance is quantified at 5% significance level.

	M10		AvS	
	Purity	NMI	Purity	NMI
Bursty HDP-LDA	0.66 ± 0.02	0.67 ± 0.01	**0.75** ± 0.03	**0.66** ± 0.01
Non-parametric ATM	0.58 ± 0.01	0.63 ± 0.00	0.69 ± 0.02	0.64 ± 0.01
CNTM w/o network	0.61 ± 0.04	0.67 ± 0.01	0.72 ± 0.03	**0.66** ± 0.01
CNTM w network	**0.67** ± 0.03	**0.69** ± 0.02	0.72 ± 0.01	**0.66** ± 0.00

Table 3: Comparison of clustering performance on the M10 and AvS dataset.

As mentioned in Section 6, we assume the ground truth classes correspond to the query categories used in creating the datasets. We evaluate the clustering performance with purity and normalised mutual information (NMI)[12] (Manning et al., 2008). Purity is a simple clustering measure which can be interpreted as the proportion of documents correctly clustered. For ground truth classes $\mathcal{S} = \{s_1, \ldots, s_J\}$ and obtained clusters $\mathcal{R} = \{r_1, \ldots, r_K\}$, the purity and NMI are computed as

$$\text{purity}(\mathcal{S}, \mathcal{R}) = \frac{1}{D} \sum_k \max_j |r_k \cap s_j| \quad , \qquad \text{NMI}(\mathcal{S}, \mathcal{R}) = \frac{2\, I(\mathcal{S}; \mathcal{R})}{H(\mathcal{S}) + H(\mathcal{R})} \quad ,$$

where $I(\mathcal{S}; \mathcal{R})$ denotes the mutual information and $H(\cdot)$ denotes the entropy:

$$I(\mathcal{S}; \mathcal{R}) = \sum_{k,\, j} \frac{|r_k \cap s_j|}{D} \log_2 \frac{D|r_k \cap s_j|}{|r_k||s_j|} \quad , \qquad H(\mathcal{R}) = -\sum_k \frac{|r_k|}{D} \log_2 \frac{|r_k|}{D} \quad .$$

The clustering results are presented in Table 3 and Table 4. 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 AI, 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 the clustering performance of CNTM with or without network is similar in Table 4. This is likely because the publications in each datasets are highly related to one another[13], and thus the citation information is not discriminating enough for clustering.

7.2. Author-merging for Semi-supervised Learning

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, thus their treatment can be problematic. In this section, we experiment on merging these authors into groups to improve document clustering. We merge authors who have authored less than η publications, to clarify, $\eta = 2$ means authors who have only a single publication are merged, while $\eta = 1$ corresponds to no merging. Additionally, we use the category labels for

12. Note that the NMI in Zhu et al. (2013) is slightly different to ours, we use the definition in Manning et al. (2008). This *penalises* our NMI result when compared against the result in Zhu et al. (2013) since our normalising term will always be equal or greater than that of Zhu et al. (2013).
13. See the list of category labels of these datasets in supplementary material.

	AI		Cora		PubMed	
	Purity	NMI	Purity	NMI	Purity	NMI
PMTLM*	N/A	0.51	N/A	0.41	N/A	0.27
CNTM w/o network	**0.51** ± 0.07	**0.67** ± 0.02	0.37 ± 0.03	**0.63** ± 0.01	**0.47** ± 0.04	**0.69** ± 0.01
CNTM w network	**0.51** ± 0.08	0.66 ± 0.02	**0.39** ± 0.03	**0.63** ± 0.02	0.46 ± 0.02	**0.69** ± 0.01

Table 4: Comparison of clustering performance of CNTM against PMTLM. The best PMTML results are chosen for comparison, from Table 2 in Zhu et al. (2013).

a semi-supervised learning. This is achieved by assigning the documents to *dummy authors* represented by the category labels, *i.e.* the authors are merged into groups based on the category labels of their publications. These groups are now considered the "authors" for the documents.

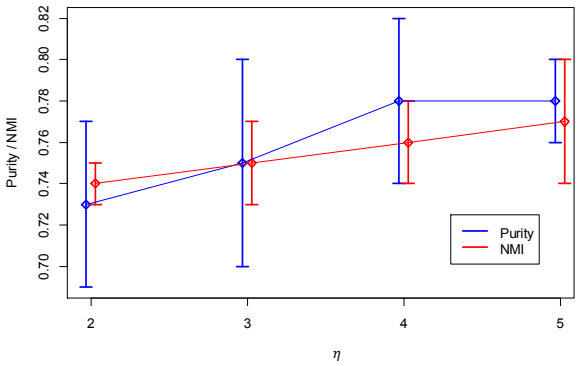

Figure 2: Plot showing the purity and NMI results for $\eta = \{2, 3, 4, 5\}$ on M10 dataset. The interval represents one standard error for estimation.

We present the clustering results for $\eta = \{2, 3, 4, 5\}$ as a plot in Figure 2 (results in table format are shown in the supplementary material). We find that increasing η generally improves the clustering performance, although the effect is not too significant for successive η. Note that if η is set to be too large, most of the author information will be replaced by the category labels, which defeats the purpose of author modelling.

7.3. Qualitative Analysis

We can obtain a summary of a text corpus from a trained CNTM, this is done by analysing the topic-word distribution ϕ. In Table 5, we display some major topics extracted from the ML dataset (M10 and AvS in supplementary material). The topics are represented by the top words, which are ordered based on ϕ_{kw}. The labels of the topics are manually assigned.

Additionally, we analyse the author-topic distributions ν to find out about authors' interests. We focus on the M10 dataset since it covers a wider area of research topics. For each author a, we determine their dominant topic from their author-topic distribution ν_a.

We display the interests of some authors in Table 6. Again, the topic labels are manually picked given the dominant topics and the corresponding top words from the topics.

Topic	Top Words
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

Table 5: Topic Summary for ML Dataset

Author	Topic	Top Words
D. Aerts	Quantum Theory	quantum, theory, quantum mechanics, classical, quantum field
Y. Bengio	Neural Network	networks, learning, recurrent neural, neural networks, models
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

Table 6: Example of authors and their topic preference learned by the CNTM.

Furthermore, we can graphically visualise the author-topics network extracted by CNTM with `Graphviz`[14]. This is detailed in the supplementary material due to space.

8. Conclusions

In this paper, we have proposed the Citation-Network Topic Model (CNTM) to jointly model research publications and their citation network. CNTM performs text modelling with a hierarchical PYP topic model and models the citations with the Poisson distribution. We also proposed a novel learning algorithm for the CNTM, which exploits the conjugacy of the Dirichlet and Multinomial distribution, allowing the sampling of the citation networks to be of similar form of 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 datasets, we demonstrate the improvement of joint topic and network modelling in terms of model fitting and clustering evaluation. Additionally, we experiment on merging authors who do not have many publications into groups of similar authors based on the query categories, giving us a semi-supervised learning. We find that clustering performance improves with the level of merging.

Future work includes learning the influences of the co-authors, utilising them for author merging and further speed up non-parametric modelling with techniques in Li et al. (2014).

Acknowledgments

NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program. The authors wish to thank CiteSeer^{X} for providing the data.

14. http://www.graphviz.org/

References

W. Buntine and M. Hutter. A Bayesian view of the Poisson-Dirichlet process. Technical Report arXiv:1007.0296v2, 2012.

W. Buntine and S. Mishra. Experiments with non-parametric topic models. In *KDD*, pages 881–890. ACM, 2014.

J. Chang and D. Blei. Hierarchical relational models for document networks. *The Annals of Applied Statistics*, 4(1):124–150, 2010.

C. Chen, L. Du, and W. Buntine. Sampling table configurations for the hierarchical Poisson-Dirichlet Process. In *ECML*, pages 296–311. Springer-Verlag, 2011.

H. Han, C. L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis. Two supervised learning approaches for name disambiguation in author citations. In *JCDL*, pages 296–305. ACM, 2004.

H. Han, H. Zha, and C. L. Giles. Name disambiguation in author citations using a K-way spectral clustering method. In *JCDL*, pages 334–343. ACM, 2005.

S. Kataria, P. Mitra, C. Caragea, and C. L. Giles. Context sensitive topic models for author influence in document networks. In *IJCAI*, pages 2274–2280. AAAI Press, 2011.

A. Li, A. Ahmed, S. Ravi, and A. Smola. Reducing the sampling complexity of topic models. In *KDD*, pages 891–900. ACM, 2014.

K. W. Lim, C. Chen, and W. Buntine. Twitter-network topic model: A full Bayesian treatment for social network and text modeling. In *NIPS Topic Model workshop*, 2013.

L. Liu, J. Tang, J. Han, M. Jiang, and S. Yang. Mining topic-level influence in heterogeneous networks. In *CIKM*, pages 199–208. ACM, 2010.

Y. Liu, A. Niculescu-Mizil, and W. Gryc. Topic-link LDA: Joint models of topic and author community. In *ICML*, pages 665–672. ACM, 2009.

M. Lui and T. Baldwin. langid.py: An off-the-shelf language identification tool. In *ACL*, pages 25–30. ACL, 2012.

C. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. ISBN 0521865719, 9780521865715.

D. Mimno and A. McCallum. Mining a digital library for influential authors. In *JCDL*, pages 105–106. ACM, 2007.

R. Nallapati, A. Ahmed, E. Xing, and W. Cohen. Joint latent topic models for text and citations. In *KDD*, pages 542–550. ACM, 2008.

J. Pitman. Some developments of the Blackwell-Macqueen urn scheme. *Lecture Notes-Monograph Series*, pages 245–267, 1996.

M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *UAI*, pages 487–494. AUAI Press, 2004.

I. Sato and H. Nakagawa. Topic models with power-law using Pitman-Yor process. In *KDD*, pages 673–682. ACM, 2010.

P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *KDD*, pages 807–816. ACM, 2009.

Y. W. Teh. A Bayesian interpretation of interpolated Kneser-Ney. Technical report, School of Computing, National University of Singapore, 2006a.

Y. W. Teh. A hierarchical Bayesian language model based on Pitman-Yor processes. In *ACL*, pages 985–992. ACL, 2006b.

Y. W. Teh and M. Jordan. Hierarchical Bayesian nonparametric models with applications. In *Bayesian Nonparametrics: Principles and Practice*. Cambridge University Press, 2010.

Y. Tu, N. Johri, D. Roth, and J. Hockenmaier. Citation author topic model in expert search. COLING, pages 1265–1273. ACL, 2010.

H. Wallach, D. Mimno, and A. McCallum. Rethinking LDA: Why priors matter. In *NIPS*, pages 1973–1981. 2009.

J. Weng, E.-P. Lim, J. Jiang, and Q. He. TwitterRank: Finding topic-sensitive influential Twitterers. In *WSDM*, pages 261–270. ACM, 2010.

Y. Zhu, X. Yan, L. Getoor, and C. Moore. Scalable text and link analysis with mixed-topic link models. In *KDD*, pages 473–481. ACM, 2013.