Mach Learn (2016) 103:185–213
DOI 10.1007/s10994-016-5554-z

Bibliographic analysis on research publications using authors, categorical labels and the citation network

Kar Wai Lim[1] · **Wray Buntine**[2]

Received: 24 March 2015 / Accepted: 23 February 2016 / Published online: 11 March 2016

Abstract Bibliographic analysis considers the author's research areas, the citation network and the 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. 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^{X}. The publication datasets are organised into three corpora, totalling to about 168k publications with about 62k authors. The queried datasets are made available online. In three publicly available corpora in addition to the queried datasets, our proposed model demonstrates an improved performance in both model fitting and document clustering, compared to several baselines. Moreover, our model allows extraction of additional useful knowledge from the corpora, such as the visualisation of the author-topics network. Additionally, we propose a simple method to incorporate supervision into topic modelling to achieve further improvement on the clustering task.

Keywords Bibliographic analysis · Topic model · Bayesian non-parametric · Author-citation network

1 Introduction

Models of bibliographic data need to consider many kinds of information. Articles are usually accompanied by metadata such as authors, publication data, categories and time. Cited papers

Editors: Hang Li, Dinh Phung, Tru Cao, Tu-Bao Ho, and Zhi-Hua Zhou.

✉ Kar Wai Lim
karwai.lim@anu.edu.au

Wray Buntine
wray.buntine@monash.edu

[1] The Australian National University (ANU) and NICTA, Canberra, Australia

[2] Monash University, Clayton, Australia

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[1] and the citation network. The model is a non-parametric extension of previous models discussed in Sect. 2. Using simple assumptions and approximations, we derive a novel algorithm that allows the probability vectors in the model to be integrated out. This yields a Markov chain Monte Carlo (MCMC) inference via discrete sampling.

As an extension of our previous work (Lim and Buntine 2014), we propose a supervised approach to improve document clustering, by making use of categorical information that is available. Our method allows the level of supervision to be adjusted through a variable, giving us a model with no supervision, semi-supervised or fully supervised. Additionally, we present a more extensive qualitative analysis of the learned topic models, and display a visualisation snapshot of the learned author-topics network. We also perform additional diagnostic tests to assess our proposed topic model. For example, we study the convergence of the proposed learning algorithm and report on the computation complexity of the algorithm.

In the next section, we discuss the related work. Sects. 3, 4 and 5 detail our topic model and its inference algorithm. We describe the datasets in Sect. 6 and report on experiments in Sect. 7. Applying our model on research publication data, we demonstrate the model's improved performance, on both model fitting and a clustering task, compared to several baselines. Additionally, in Sect. 8, we qualitatively analyse the inference results produced by our model. 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 the topic model in Sect. 9 and conclude the paper in Sect. 10.

2 Related work

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 *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, also known as the two-parameter Poisson Dirichlet process) (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).

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.

[1] Abstract and publication title.

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 network[2] 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.

3 Supervised Citation Network Topic Model

In our previous work (Lim and Buntine 2014), we proposed the Citation Network Topic Model (CNTM) that jointly models the *text*, *authors*, and the *citation network* of research publications (documents). The CNTM allows us to both model the authors and text better by exploiting the correlation between the authors and their research topics. However, the benefit of the above modelling is not realised when the author information is simply missing from the data. This could be due to error in data collection (e.g. metadata not properly formatted), or even simply that the author information is lost during preprocessing.

In this section, we propose an extension of the CNTM that remedies the above issue, by making use of additional metadata that is available. For example, the metadata could be the research areas or keywords associated with the publications, which are usually provided by the authors during the publication submission. However, this information might not always be reliable as it is not standardised across different publishers or conferences. In this paper, rather than using the mentioned metadata, we will instead incorporate the categorical labels that were previously used as ground truth for evaluation. As such, our extension gives rise to a supervised model, which we will call the Supervised Citation Network Topic Model (SCNTM).

We first describe the topic model part of SCNTM for which the citations are not considered, it will be used for comparison later in Sect. 7. We then complete the SCNTM with the discussion on its network component. The full graphical model for SCNTM is displayed in Fig. 1.

To clarify the notations used in this paper, *variables that are without subscript represent a collection of variables of the same notation*. For instance, w_d represents 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.

3.1 Hierarchical Pitman–Yor topic model

The SCNTM uses both the *Griffiths–Engen–McCloskey* (GEM) distribution (Pitman 1996) and the *Pitman–Yor process* (PYP) (Teh 2006b) to generate probability vectors. Both the GEM distribution and the PYP are parameterised by a *discount* parameter α and a *concentration* parameter β. The PYP is additionally parameterised by a *base distribution* H, which is also

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

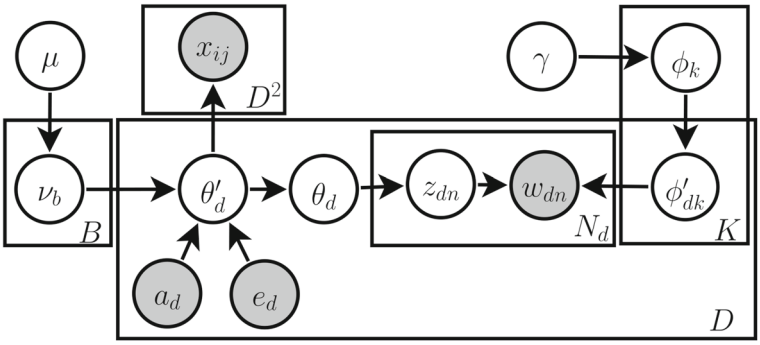

Fig. 1 Graphical model for SCNTM. The *box* on the *top left* with D^2 entries is the citation network on documents represented as a Boolean matrix. The remainder is a non-parametric hierarchical PYP topic model where the labelled categories and authors are captured by the topic vectors ν. The topic vectors ν influence the D documents' topic vectors θ' and θ based on the observed authors a or categories e. The latent topics and associated words are represented by the variables z and w. The K topics, shown in the *top right*, have bursty modelling following Buntine and Mishra (2014)

the mean of the PYP when it can be represented by a probability vector. Note that the base distribution can also be a PYP. This gives rise to the hierarchical Pitman–Yor process (HPYP).

In modelling authorship, the SCNTM 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. 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 generative process of the topic model component of the SCNTM 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, and also for the category-topic distributions ν_e for each category e:

$$\mu \sim \text{GEM}(\alpha^{\mu}, \beta^{\mu}), \tag{1}$$

$$\nu_a \mid \mu \sim \text{PYP}(\alpha^{\nu_a}, \beta^{\nu_a}, \mu), \quad a \in \mathcal{A}. \tag{2}$$

$$\nu_e \mid \mu \sim \text{PYP}(\alpha^{\nu_e}, \beta^{\nu_e}, \mu), \quad e \in \mathcal{E}. \tag{3}$$

Here, $\mathcal{A}$ represents the set of all authors while $\mathcal{E}$ denotes the set of all categorical labels in the text corpus. Note we have used the same symbol (ν) for both the author-topic distributions and the category-topic distributions.

We introduce a parameter η called the *author threshold* which controls the level of supervision used by SCNTM. We say an author a is significant if the author has produced more than or equal to η publications, i.e.

$$\text{significance}(a) = \begin{cases} 1 & \text{if } \sum_d I(a_d = a) \geq \eta \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Here, a_d represents the author for document d, and $I(\triangle)$ is the indicator function that evaluates to 1 if $\triangle$ is true, else 0.

Next, for each document d in a publication collection of size D, we sample the document-topic prior θ'_d from ν_{a_d} or ν_{e_d} depending on whether the author a_d for the document is significant:

$$\theta'_d \mid a_d, e_d, \nu \sim \begin{cases} \text{PYP}\left(\alpha^{\theta'_d}, \beta^{\theta'_d}, \nu_{a_d}\right) & \text{if significance}(a_d) = 1 \\ \text{PYP}\left(\alpha^{\theta'_d}, \beta^{\theta'_d}, \nu_{e_d}\right) & \text{otherwise,} \end{cases} \quad d = 1, \ldots, D, \tag{5}$$

where e_d is the categorical label associated with document d. For the sake of notational simplicity, we introduce a variable b to capture both the author and the category. We let b takes the value of $1, \ldots, A$ for each author in $\mathcal{A}$, and let b takes the value of $(A+1), \ldots, B$ for the categories in $\mathcal{E}$. Note that $B = |\mathcal{A}| + |\mathcal{E}|$. Thus, we can also write the distribution of θ'_d as

$$\theta'_d \mid \nu_b \sim \text{PYP}(\alpha^{\theta'_d}, \beta^{\theta'_d}, \nu_b) \quad d = 1, \ldots, D, \tag{6}$$

where $b = a_d$ if significance$(a_d) = 1$, else $b = e_d$.

By modelling this way, we are able to handle missing authors and incorporate supervision into the SCNTM. For example, choosing $\eta = 1$ allows us to make use of the categorical information for documents that have no valid author. Alternatively, we could select a higher η, this smooths out the document-topic distributions for documents that are written by authors who have authored only a small number of publications. This treatment leads to a better clustering result as these authors are usually not discriminative enough for prediction. On the extreme, we can set $\eta = \infty$ to achieve full supervision. We note that the SCNTM reverts to the CNTM when $\eta = 0$, in this case the model is not supervised.

We then sample the document-topic distribution θ_d given θ'_d:

$$\theta_d \mid \theta'_d \sim \text{PYP}(\alpha^{\theta_d}, \beta^{\theta_d}, \theta'_d), \quad d = 1, \ldots, D. \tag{7}$$

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 take into account the relatively low amount of citations compared to the text. The technical details on the effect of such modelling is presented in Sect. 9.2.

For the vocabulary side, we generate a background word distribution γ given H^γ, a discrete uniform vector of length $|\mathcal{V}|$, i.e. $H^\gamma = (\ldots, \frac{1}{|\mathcal{V}|}, \ldots)$. $\mathcal{V}$ is 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:

$$\gamma \sim \text{PYP}(\alpha^\gamma, \beta^\gamma, H^\gamma), \tag{8}$$

$$\phi_k \mid \gamma \sim \text{PYP}(\alpha^{\phi_k}, \beta^{\phi_k}, \gamma), \quad k = 1, \ldots, K. \tag{9}$$

Modelling word burstiness (Buntine and Mishra 2014) is important since words in a document are likely to repeat in the document. The same applies to publication abstract, as shown in Sect. 6. To address this property, we make the topics bursty so each document only focuses on a subset of words in the topic. This is achieved by defining the document-specific topic-word distribution ϕ'_{dk} for each topic k in document d as:

$$\phi'_{dk} \mid \phi_k \sim \text{PYP}(\alpha^{\phi'_{dk}}, \beta^{\phi'_{dk}}, \phi_k), \quad d = 1, \ldots, D, \ k = 1, \ldots, K. \tag{10}$$

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}:

$$z_{dn} \mid \theta_d \sim \text{Discrete}(\theta_d), \tag{11}$$

$$w_{dn} \mid z_{dn}, \phi'_d \sim \text{Discrete}(\phi'_{dz_{dn}}), \quad d = 1, \ldots, D, \ n = 1, \ldots, N_d. \tag{12}$$

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

In the next section, we describe the modelling of the citation network accompanying a publication collections. This completes the SCNTM.

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 θ' 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}:

$$\begin{aligned} x_{ij} \mid \lambda_{ij} &\sim \text{Poisson}(\lambda_{ij}), \\ \lambda_{ij} &= \lambda_i^+ \lambda_j^- \textstyle\sum_k \lambda_k^T \theta'_{ik} \theta'_{jk}, \quad i = 1, \ldots, D, \quad j = 1, \ldots, D. \end{aligned} \tag{13}$$

Here, λ_i^+ is the propensity of document i to cite and λ_j^- represents the popularity of cited document j, while 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. Due to the limitation of the data, the x_{ij} can only be 0 or 1, i.e. it is a Boolean variable. Nevertheless, the Poisson distribution is used instead of a Bernoulli distribution because it leads to dramatically reduced complexity in analysis (Zhu et al. 2013). 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 SCNTM in Table 1.

4 Model representation and posterior likelihood

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

4.1 Modelling with hierarchical PYPs

The key to efficient 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).

Table 1 List of variables for the Supervised Citation Network Topic Model (SCNTM)

Variable	Name	Description
z_{dn}	Topic	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
a_d	Author	Author for document d
e_d	Category	Category label for document d
ϕ'_{dk}	Document-topic-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_b	Author/category-topic distribution	Probability distribution in generating topics for author or category b
γ	Global word distribution	Word prior for ϕ_k
μ	Global topic distribution	Topic prior for ν_b
$\alpha^{\mathcal{N}}$	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 $\mathcal{N}$
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

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* $c^{\mathcal{N}} = (\ldots, c_k^{\mathcal{N}}, \ldots)$ and the *table counts* $t^{\mathcal{N}} = (\ldots, t_k^{\mathcal{N}}, \ldots)$. 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 $\mathcal{N}$). Thus $t_k^{\mathcal{N}} \leq c_k^{\mathcal{N}}$, and $t_k^{\mathcal{N}} = 0$ if and only if $c_k^{\mathcal{N}} = 0$ since the counts are non-negative. 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 $f^*(\mathcal{N})$, in terms of $c^{\mathcal{N}}$ and $t^{\mathcal{N}}$, is given as

$$f^*(\mathcal{N}) = \frac{\left(\beta^{\mathcal{N}} | \alpha^{\mathcal{N}}\right)_{T^{\mathcal{N}}}}{\left(\beta^{\mathcal{N}}\right)_{C^{\mathcal{N}}}} \prod_k S^{c_k^{\mathcal{N}}}_{t_k^{\mathcal{N}}, \alpha^{\mathcal{N}}}, \quad \text{for } \mathcal{N} \sim \text{PYP}\left(\alpha^{\mathcal{N}}, \beta^{\mathcal{N}}, \mathcal{P}\right). \tag{14}$$

$S^x_{y,\alpha}$ is the generalised Stirling number that is easily tabulated; 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 Eq. 14 directly led to poor performance. The problem was that sampling an assignment to a latent variable, say

moving a customer from group k to k' (so $c_k^{\mathcal{N}}$ decreases by 1 and $c_{k'}^{\mathcal{N}}$ increases by 1), the potential effect on $t_k^{\mathcal{N}}$ and $t_{k'}^{\mathcal{N}}$ could not immediately be measured. Whereas, the hierarchical CRP automatically included table configurations in its sampling process and thus included the influence of the hierarchy in the sampling. Thus sampling directly with Eq. 14 lead to comparatively poor mixing. As a solution, Chen et al. (2011) develop a collapsed version of the hierarchical CRP following the well known practice of Rao-Blackwellisation of sampling schemes (Casella and Robert 1996), which, while not being as fast per step, it has two distinct advantages, (1) it requires no dynamic memory and (2) the sampling has significantly lower variance so converges much faster. This has empirically been shown to lead to better mixing of the samplers (Chen et al. 2011) and has been confirmed on different complex topic models (Buntine and Mishra 2014).

The technique for collapsing the hierarchical CRP uses Eq. 14 but the counts $(c^{\mathcal{N}}, t^{\mathcal{N}})$ are now derived variables. They are derived from Boolean variables associated with each data point. The technique comprises the following conceptual steps: (1) add Boolean indicators u_{dn} to the data (z_{dn}, w_{dn}) from which the counts $c^{\mathcal{N}}$ and $t^{\mathcal{N}}$ can be derived, (2) modify the marginalised posterior accordingly, and (3) derive a sampler for the model.

4.1.1 Adding Boolean indicators

We first consider $c_k^{\theta_d}$, which has a "+1" contributed to for every $z_{dn} = k$ in document d, hence $c_k^{\theta_d} = \sum_n I(z_{dn} = k)$. We now introduce a new Bernoulli *indicator variable* $u_{dn}^{\theta_d}$ associated with z_{dn}, which is "on" (or 1) when the data z_{dn} also contributed a "+1" to $t_k^{\theta_d}$. Note that $t_k^{\theta_d} \le c_k^{\theta_d}$, so every data contributing a "+1" to $c_k^{\theta_d}$ may or may not contribute a "+1" to $t_k^{\theta_d}$. The result is that one derives $t_k^{\theta_d} = \sum_n I(z_{dn} = k)\, I(u_{dn}^{\theta_d} = 1)$.

Now consider the parent of θ_d, which is θ'_d. Its customer count is derived as $c_k^{\theta'_d} = t_k^{\theta_d}$. Its table count $t_k^{\theta'_d}$ can now be treated similarly. Those data z_{dn} that contribute a "+1" to $t_k^{\theta_d}$ (and thus $c_k^{\theta'_d}$) have a new Bernoulli indicator variable $u_{dn}^{\theta'_d}$, which is used to derive $t_k^{\theta'_d} = \sum_n I(z_{dn} = k)\, I(u_{dn}^{\theta'_d} = 1)$, similar as before. Note that if $u_{dn}^{\theta'_d} = 1$ then necessarily $u_{dn}^{\theta_d} = 1$.

Similarly, one can define Boolean indicators for μ, ν_b, ϕ', ϕ, and γ to have a full suite from which all the counts $c^{\mathcal{N}}$ and $t^{\mathcal{N}}$ are now derived. We denote $u_{dn} = \{u_{dn}^{\theta_d}, u_{dn}^{\theta'_d}, u_{dn}^{\nu_b}, u_{dn}^{\mu}, u_{dn}^{\phi'_d}, u_{dn}^{\phi_d}, u_{dn}^{\gamma}\}$ as the collection of the Boolean indicators for data (z_{dn}, w_{dn}).

4.1.2 Probability of Boolean indicators

By symmetry, if there are $t_k^{\mathcal{N}}$ Boolean indicators "on" (out of $c_k^{\mathcal{N}}$), we are indifferent as to which is on. Thus the indicator variable $u_{dn}^{\mathcal{N}}$ is not stored, that is, we simply "forget" who contributed a table count and re-sample $u_{dn}^{\mathcal{N}}$ as needed:

$$p\left(u_{dn}^{\mathcal{N}} = 1\right) = t_k^{\mathcal{N}} / c_k^{\mathcal{N}}, \quad p\left(u_{dn}^{\mathcal{N}} = 0\right) = 1 - t_k^{\mathcal{N}} / c_k^{\mathcal{N}}. \tag{15}$$

Moreover, this means that the marginalised likelihood $f^*(\mathcal{N})$ of Eq. 14 is extended to include the probability of $u^{\mathcal{N}}$, which is written in terms of $c^{\mathcal{N}}$, $t^{\mathcal{N}}$ and $u^{\mathcal{N}}$ as:

$$f(\mathcal{N}) = f^*(\mathcal{N})\; p\left(u^{\mathcal{N}} \,\middle|\, c^{\mathcal{N}}, t^{\mathcal{N}}\right) = f^*(\mathcal{N}) \prod_k \binom{c_k^{\mathcal{N}}}{t_k^{\mathcal{N}}}^{-1}. \tag{16}$$

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, $\mathbf{Z} = \{z_{11}, \ldots, z_{DN_D}\}$ denotes the set of all topic assignments. Variables $\mathbf{W}$, $\mathbf{T}$, $\mathbf{C}$ and $\mathbf{U}$ are similarly defined, that is, they denote the set of all words, table counts, customer counts, and Boolean indicators 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 given in Eq. 16—as $p(\mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}, \mathbf{U} \mid \zeta) \propto$

$$f(\mu)\left(\prod_{b=1}^{B} f(\nu_b)\right)\left(\prod_{d=1}^{D} f(\theta'_d)\, f(\theta_d) \prod_{k=1}^{K} f(\phi'_{dk})\right)\left(\prod_{k=1}^{K} f(\phi_k)\right) f(\gamma)\left(\prod_{v}\left(\frac{1}{|\mathcal{V}|}\right)^{t_v^{\gamma}}\right). \tag{17}$$

Note that the last term in Eq. 17 corresponds to the parent probability vector of γ (see Sect. 3.1), and v indexes the unique word tokens in vocabulary set $\mathcal{V}$. Note that the extra terms for $\mathbf{U}$ are simply derived using Eq. 16 and not stored in the model. So in the discussions below we will usually represent $\mathbf{U}$ implicitly by $\mathbf{T}$ and $\mathbf{C}$, and introduce the $\mathbf{U}$ when explicitly needed.

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 $\mathcal{N}$ can be estimated from its posterior mean given the counts and parent probability vector $\mathcal{P}$:

$$\hat{\mathcal{N}} = \left(\ldots, \frac{\left(\alpha^{\mathcal{N}} T^{\mathcal{N}} + \beta^{\mathcal{N}}\right)\mathcal{P}_k + c_k^{\mathcal{N}} - \alpha^{\mathcal{N}} T_k^{\mathcal{N}}}{\beta^{\mathcal{N}} + C^{\mathcal{N}}}, \ldots\right). \tag{18}$$

4.3 Likelihood for the Citation Network Poisson Model

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

$$p(x_{ij} \mid \lambda, \theta) = \frac{\lambda_{ij}^{x_{ij}}}{x_{ij}!\, e^{\lambda_{ij}}} \approx \left(\lambda_i^{+}\lambda_j^{-} \sum_k \lambda_k^T \theta'_{ik}\theta'_{jk}\right)^{x_{ij}} \exp\left(-\lambda_i^{+}\lambda_j^{-} \sum_k \lambda_k^T \theta'_{ik}\theta'_{jk}\right). \tag{19}$$

Note that the term $x_{ij}!$ is dropped in Eq. 19 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 $\mathbf{X} = \{x_{11}, \ldots, x_{DD}\}$ can be written as $p(\mathbf{X} \mid \lambda, \theta') =$

$$\left(\prod_i (\lambda_i^{+})^{g_i^{+}} (\lambda_i^{-})^{g_i^{-}}\right)\prod_{ij}\left(\sum_k \lambda_k^T \theta'_{ik}\theta'_{jk}\right)^{x_{ij}} \exp\left(-\sum_{ijk} \lambda_i^{+}\lambda_j^{-}\lambda_k^T \theta'_{ik}\theta'_{jk}\right), \tag{20}$$

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 that $x_{ii} = 1$ for all documents i, 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 Eq. 20, which leads to a cleaner learning algorithm that utilises an efficient caching. Note that if we do not define x_{ii}, we have to explicitly consider the case when $i = j$ in Eq. 20 which results in messier summation and products.

Note the likelihood in Eq. 20 contains the document-topic distribution θ' in vector form. This is problematic as performing inference with the likelihood requires the probability vectors θ', ν and μ to be stored explicitly (instead of counts as discussed in Sect. 4.1). To

overcome this issue, we propose a novel representation that allows the probability vectors to remain integrated out. Such representation also leads to an efficient sampling algorithm for the citation network, as we will see in Sect. 5.

We introduce an *auxiliary variable* y_{ij}, named the *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 Eq. 13, we model the citing topic y_{ij} as jointly Poisson with x_{ij}:

$$x_{ij}, y_{ij} = k \mid \lambda, \theta' \sim \text{Poisson}\left(\lambda_i^+ \lambda_j^- \lambda_k^T \theta'_{ik} \theta'_{jk}\right). \tag{21}$$

Incorporating $\mathbf{Y}$, the set of all y_{ij}, we rewrite the citation network likelihood as $p(\mathbf{X}, \mathbf{Y} | \lambda, \theta') \propto$

$$\prod_i (\lambda_i^+)^{g_i^+} (\lambda_i^-)^{g_i^-} \prod_k \left(\lambda_k^T\right)^{\frac{1}{2}\sum_i h_{ik}} \prod_{ik} {\theta'_{ik}}^{h_{ik}} \exp\left(-\sum_{ij} \lambda_i^+ \lambda_j^- \lambda_{y_{ij}}^T \theta'_{i y_{ij}} \theta'_{j y_{ij}}\right), \tag{22}$$

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 $p(\mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}, \mathbf{U} \mid \zeta)$ 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 to 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 the delta method (Oehlert 1992) to approximate $\int q(\theta) \exp(-g(\theta))\, \mathrm{d}\theta \approx \exp(-g(\hat{\theta})) \int q(\theta)\, \mathrm{d}\theta$, where $\hat{\theta}$ is the expected value according to a distribution proportional to $q(\theta)$. This approximation is reasonable as long as the terms in the exponential are small (see "Appendix 1"). The approximate full posterior of SCNTM can then be written as $p(\mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}^+, \mathbf{U}, \mathbf{X}, \mathbf{Y} \mid \lambda, \zeta) \approx$

$$p(\mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}^+, \mathbf{U} \mid \zeta) \prod_i \left(\lambda_i^+\right)^{g_i^+} \left(\lambda_i^-\right)^{g_i^-} \prod_k \left(\lambda_k^T\right)^{g_k^T} \exp\left(-\sum_{ij} \lambda_i^+ \lambda_j^- \lambda_{y_{ij}}^T \hat{\theta}'_{i y_{ij}} \hat{\theta}'_{j y_{ij}}\right), \tag{23}$$

where $g_k^T = \frac{1}{2}\sum_i h_{ik}$. We note that $p(\mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}^+, \mathbf{U} \mid \zeta)$ is the same as Eq. 17 but now with $\mathbf{C}^+$ instead of $\mathbf{C}$.

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 SCNTM. We first describe the sampler for the topic model and then for the citation network. The full inference procedure is performed by alternating between the two samplers. Finally, we outline the hyperparameter samplers that are used to estimate the hyperparameters automatically.

5.1 Sampling for the hierarchical PYP topic model

To sample the words' topic $\mathbf{Z}$ and the associated counts $\mathbf{T}$ and $\mathbf{C}$ in the SCNTM, we design a Metropolis–Hastings (MH) algorithm based on the collapsed Gibbs sampler designed for the PYP (Chen et al. 2011). The concept of the MH sampler is analogous to LDA, which consists of (1) decrementing the counts associated with a word, (2) sampling the respective new topic assignment for the word, and (3) incrementing the associated counts. However, our sampler is more complicated than LDA. In particular, we have to consider the indicators $u_{dn}^{\mathcal{N}}$ described in Sect. 4.1 operating on the hierarchy of PYPs. Our MH sampler consists of two steps. First we sample the latent topic z_{dn} associated with the word w_{dn}. We then sample the customer counts $\mathbf{C}$ and table counts $\mathbf{T}$.

The sampler proceeds by considering the latent variables associated with a given word w_{dn}. First, we decrement the counts associated with the word w_{dn} and the latent topic z_{dn}. This is achieved by sampling the suite of indicators u_{dn} according to Eq. 15 and decrementing the relevant customer counts and table counts. For example, we decrement $c_{z_{dn}}^{\theta_d}$ by 1 if $u_{dn}^{\theta_d} = 1$. After decrementing, we apply a Gibbs sampler to sample a new topic z_{dn} from its conditional posterior distribution, given as $p(z_{dn}^{\text{new}} \mid \mathbf{Z}^{-dn}, \mathbf{W}, \mathbf{T}^{-dn}, \mathbf{C}^{+-dn}, \mathbf{U}^{-dn}, \zeta) =$

$$\sum_{u_{dn}} p\left(z_{dn}^{\text{new}}, u_{dn} \mid \mathbf{Z}^{-dn}, \mathbf{W}, \mathbf{T}^{-dn}, \mathbf{C}^{+-dn}, \mathbf{U}^{-dn}, \zeta\right). \tag{24}$$

Note that the joint distribution in Eq. 24 can be written as the ratio of the likelihood for the topic model (Eq. 17):

$$\frac{p(\mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}^{+}, \mathbf{U} \mid \zeta)}{p(\mathbf{Z}^{-dn}, \mathbf{W}, \mathbf{T}^{-dn}, \mathbf{C}^{+-dn}, \mathbf{U}^{-dn} \mid \zeta)}. \tag{25}$$

Here, the superscript $\mathcal{O}^{-dn}$ 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. Additionally, we use the superscripts $\mathcal{O}^{\text{new}}$ and $\mathcal{O}^{\text{old}}$ to denote the proposed sample and the old value respectively. The modularised likelihood of Eq. 17 allows the conditional posterior (Eq. 24) to be computed easily, since it simplifies to ratios of likelihood $f(\cdot)$, which simplifies further since 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 $S_{x+1,\alpha}^{y+1}/S_{x,\alpha}^{y}$, can be computed quickly via caching (Buntine and Hutter 2012).

Next, we proceed to sample the relevant customer counts and table counts given the new $z_{dn} = k$. We propose an MH algorithm for this. We define the proposal distribution for the new customer counts and table counts as

$$q\left(\mathbf{T}^{\text{new}}, \mathbf{C}^{+\text{new}} \,\middle|\, \mathbf{Z}, \mathbf{W}, \mathbf{T}^{-dn}, \mathbf{C}^{+-dn}, \zeta\right) \propto \frac{p\left(\mathbf{Z}, \mathbf{W}, \mathbf{T}^{\text{new}}, \mathbf{C}^{+\text{new}}, \mathbf{U}^{\text{new}} \,\middle|\, \zeta\right)}{p\left(\mathbf{Z}, \mathbf{W}, \mathbf{T}^{-dn}, \mathbf{C}^{+-dn}, \mathbf{U}^{-dn} \,\middle|\, \zeta\right)} \tag{26}$$

where

$$\begin{aligned} p(\mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}^{+}, \mathbf{U} \mid \zeta) \propto f(\mu) \left(\prod_{b=1}^{B} f(\nu_b)\right) \left(\prod_{d=1}^{D} f(\theta_d') \, f(\theta_d) \prod_{k=1}^{K} f(\phi_{dk}')\right) \\ \times \left(\prod_{k=1}^{K} f(\phi_k)\right) f(\gamma) \left(\prod_{v} \left(\frac{1}{|\mathcal{V}|}\right)^{t_v^{\gamma}}\right). \end{aligned} \tag{27}$$

Here, the potential sample space for $\mathbf{T}^{\text{new}}$ and $\mathbf{C}^{\text{new}}$ are restricted to just $t_k + i$ and $c_k + i$ where i is either 0 or 1. Doing so allows us to avoid considering the exponentially many

possibilities of $\mathbf{T}$ and $\mathbf{C}$. The acceptance probability associated with the newly sampled $\mathbf{T}^{\text{new}}$ and $\mathbf{C}^{\text{new}}$ is

$$A = \frac{p\left(\mathbf{Z}, \mathbf{W}, \mathbf{T}^{\text{new}}, \mathbf{C}^{+^{\text{new}}}, \mathbf{U}^{\text{new}} \mid \zeta\right)}{p\left(\mathbf{Z}, \mathbf{W}, \mathbf{T}^{old}, \mathbf{C}^{+^{old}}, \mathbf{U}^{old} \mid \zeta\right)} \cdot \frac{q\left(\mathbf{T}^{\text{old}}, \mathbf{C}^{+^{\text{old}}} \mid \mathbf{Z}, \mathbf{W}, \mathbf{T}^{-dn}, \mathbf{C}^{+^{-dn}}\zeta\right)}{q\left(\mathbf{T}^{\text{new}}, \mathbf{C}^{+^{\text{new}}} \mid \mathbf{Z}, \mathbf{W}, \mathbf{T}^{-dn}, \mathbf{C}^{+^{-dn}}\zeta\right)} = 1. \tag{28}$$

Thus we always accept the proposed sample.[3] Note that since μ is GEM distributed, incrementing t_k^{μ} is equivalent to sampling a *new* topic, i.e. the number of topics increases by 1.

5.2 Sampling for the citation network

For the citation network, we propose another MH algorithm. The MH algorithm can be summarised in three steps: (1) estimate the document topic prior θ', (2) propose a new citing topic y_{ij}, and (3) accept or reject the proposed y_{ij} following an MH scheme. Note that the MH algorithm is similar to the sampler for the topic model, 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. 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.

We describe our MH algorithm for the citation network as follows. First, for each document d, we estimate the expected document-topic prior $\hat{\theta}'_d$ from Eq. 18. Then, for each document pair (i, j) where $x_{ij} = 1$, we decrement the network counts associated with x_{ij}, and re-sample y_{ij} with a proposal distribution derived from Eq. 21:

$$p\left(y_{ij}^{\text{new}} = k \mid \hat{\theta}'_i, \hat{\theta}'_j\right) \propto \lambda_k^T \hat{\theta}'_{ik} \hat{\theta}'_{jk} \exp\left(-\lambda_i^+ \lambda_j^- \lambda_k^T \hat{\theta}'_{ik} \hat{\theta}'_{jk}\right), \tag{29}$$

which can be further simplified since the terms inside the exponential are very small, hence the exp 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 new citing topic y_{ij}^{new}. So we ignore the exponential term and let

$$p(y_{ij}^{\text{new}} = k \mid \hat{\theta}'_i, \hat{\theta}'_j) \propto \lambda_k^T \hat{\theta}'_{ik} \hat{\theta}'_{jk}. \tag{30}$$

We compute the acceptance probability A 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 $\hat{\theta}'^{\text{new}}$):

$$A = \frac{\exp\left(-\sum_{ijk} \lambda_i^+ \lambda_j^- \lambda_k^T \hat{\theta}'^{\text{new}}_{ik} \hat{\theta}'^{\text{new}}_{jk}\right)}{\exp\left(-\sum_{ijk} \lambda_i^+ \lambda_j^- \lambda_k^T \hat{\theta}'^{\text{old}}_{ik} \hat{\theta}'^{\text{old}}_{jk}\right)} \frac{p(\mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}^{+^{\text{new}}}, \mathbf{U} \mid \zeta)}{p(\mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{C}^{+^{\text{old}}}, \mathbf{U} \mid \zeta)} \times \frac{\lambda_{y^*}^T \hat{\theta}'^{\text{new}}_{iy^*} \hat{\theta}'^{\text{new}}_{jy^*}}{\lambda_{y'}^T \theta'^{\text{old}}_{iy'} \theta'^{\text{old}}_{jy'}} \frac{\sum_k \lambda_k^T \hat{\theta}'^{\text{old}}_{ik} \hat{\theta}'^{\text{old}}_{jk}}{\sum_k \lambda_k^T \hat{\theta}'^{\text{new}}_{ik} \hat{\theta}'^{\text{new}}_{jk}}. \tag{31}$$

[3] The algorithm is named MH algorithm instead of Gibbs sampling due to the fact that the sample space for the counts is restricted and thus we are not sampling from the posterior directly.

Note that we have abused the notations i and j in the above equation, 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 make things less confusing.

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

5.3 Hyperparameter sampling

Hyperparameter sampling for the priors is 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), 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.

Here we outline the procedure to sample the concentration parameter $\beta^{\mathcal{N}}$ of a PYP distributed variable $\mathcal{N}$, using an auxiliary variable sampler. Assuming each $\beta^{\mathcal{N}}$ has a Gamma distributed hyperprior with shape τ_0 and rate τ_1, we first sample the auxiliary variables ξ and ψ_j for $j \in \{0, T^{\mathcal{N}} - 1\}$:

$$\xi \mid \beta^{\mathcal{N}} \sim \text{Beta}(C^{\mathcal{N}}, \beta^{\mathcal{N}}), \quad \psi_j \mid \alpha^{\mathcal{N}}, \beta^{\mathcal{N}} \sim \text{Bernoulli}\left(\frac{\beta^{\mathcal{N}}}{\beta^{\mathcal{N}} + j\alpha^{\mathcal{N}}}\right). \tag{32}$$

We then sample a new $\beta'^{\mathcal{N}}$ from the following conditional posterior given the auxiliary variables:

$$\beta'^{\mathcal{N}} \mid \xi, \psi \sim \text{Gamma}\left(\tau_0 + \textstyle\sum_j \psi_j, \tau_1 - \log(1 - \xi)\right). \tag{33}$$

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.

$$(\lambda_i^+ \mid \mathbf{X}, \lambda^-, \lambda^T\theta') \sim \text{Gamma}\left(\epsilon_0 + g_i^+, \epsilon_1 + \textstyle\sum_k \lambda_k^T \theta'_{ik} \sum_j \lambda_j^- \theta'_{jk}\right), \tag{34}$$

$$(\lambda_i^- \mid \mathbf{X}, \lambda^+, \lambda^T\theta') \sim \text{Gamma}\left(\epsilon_0 + g_i^-, \epsilon_1 + \textstyle\sum_k \lambda_k^T \theta'_{ik} \sum_j \lambda_j^+ \theta'_{jk}\right), \tag{35}$$

$$(\lambda_k^T \mid \mathbf{X}, \mathbf{Y}, \lambda^+, \lambda^-, \theta') \sim \text{Gamma}\left(\epsilon_0 + \tfrac{1}{2}\textstyle\sum_i h_{ik}, \epsilon_1 + \lambda_k^T \left(\sum_j \lambda_j^+ \theta'_{jk}\right)\left(\sum_j \lambda_j^- \theta'_{jk}\right)\right). \tag{36}$$

We apply vague priors to the hyperpriors by setting $\tau_0 = \tau_1 = \epsilon_0 = \epsilon_1 = 1$.

Before we proceed with the next section on the datasets used in the paper, we summarise the full inference algorithm for the SCNTM 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, which are queried from CiteSeer$^{\text{X}}$ using the keywords from Microsoft Academic Search.[5] The ML

[4] http://citeseerx.ist.psu.edu/.

[5] http://academic.research.microsoft.com/.

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. Sample a new topic z_{dn} with its conditional posterior in Eq. 24
   iii. Sample the counts **T** and **C** with the proposal distribution in Eq. 26
3. For each citation $x_{ij} = 1$:
   i. Decrement the network counts associated with x_{ij} and y_{ij}
   ii. Sample a new citing topic y_{ij} with the proposal distribution in Eq. 30
   iii. Accept or reject the sampled y_{ij} with the acceptance probability in Eq. 31
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

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 science disciplines. Arts publications are made of *history* and *religion* publications, while the science publications contain *physics*, *chemistry* and *biology* research. This dataset consists of 18,720 publications and is named Arts versus Science (AvS) in this paper. These queried datasets are made available online.[6]

The keywords used to create the datasets are obtained from Microsoft Academic Search, and are listed in "Appendix 2". For the clustering evaluation in Sect. 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 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 Sect. 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)[7] 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, i.e. 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 semantics. 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 3"). We present a summary of the datasets in Table 2 and their respective categorical labels in Table 3.

[6] http://karwai.weebly.com/publications.html.

[7] http://linqs.cs.umd.edu/projects/projects/lbc/.

Table 2 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 average percentage of unique words repeated in a document

Datasets	Publications	Citations	Authors	Vocabulary	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

Author information is not available in the last three datasets

Table 3 Categories of the datasets

Datasets	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

6.1 Data noise removal

Here, we briefly discuss the steps taken to reduce the corrupted entries in 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 remove the

invalid authors using a list of exclusion words. The list of exclusion words is presented in "Appendix 4".

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. Then, we use `LingPipe` (Carpenter 2004)[8] to extract the significant phrases from the respective datasets. We refer the readers to the online tutorial[9] for details. 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` (McCallum 2002).[10] 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 Sect. 7.

7 Experiments and results

In this section, we describe experiments that compare the SCNTM 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). 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.

7.1 Experimental 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 SCNTM (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 non-parametric topic modelling. To prevent the learned topics from being too fine-grained, we set a limit to the maximum number of topics that can be learned. In particular, we have the number of topics cap at 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.

[8] http://alias-i.com/lingpipe/.

[9] http://alias-i.com/lingpipe/demos/tutorial/interestingPhrases/read-me.html.

[10] http://mallet.cs.umass.edu/.

When training the topic models, we run the inference algorithm for 2,000 iterations. For the SCNTM, 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 Sect. 9.1). We repeat each experiment five times to reduce the estimation error of the evaluation measures.

7.2 Estimating the test documents' topic distributions

The topic distribution θ' on the test documents is required to perform 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 *independently* estimate the topic assignment z_{dn} of word w_{dn} by sampling from its predictive posterior distribution given the learned topic distributions ν and topic-word distributions ϕ:

$$p(z_{dn} = k \mid w_{dn}, \nu, \phi) \propto \nu_{bk}\, \phi_{k w_{dn}}, \tag{37}$$

where $b = a_d$ if significance$(a_d) = 1$, else $b = e_d$. Note that the intermediate distributions ϕ' are integrated out (see "Appendix 5").

We then build the customer counts c^{θ_d} 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 Eq. 18.

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 $\theta'^{(s)}_d$. We adopt a Monte Carlo approach by generating $R = 500$ samples of $\theta'^{(s)}_d$, and calculate the Monte Carlo estimate of θ'_d:

$$\hat{\theta}'_d = \frac{\sum_s \theta'^{(s)}_d}{R}. \tag{38}$$

7.3 Goodness-of-fit test

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 $\mathbf{W}$ given the model, so the lower the better:

$$\text{Perplexity}(\mathbf{W}) = \exp\left(-\frac{\sum_{d=1}^{D}\sum_{n=1}^{N_d} \log p(w_{dn} \mid \theta'_d, \phi)}{\sum_{d=1}^{D} N_d}\right), \tag{39}$$

Table 4 Perplexity for the train and test documents for all datasets, lower perplexity is better

Models	Perplexity			
	Train	Test	Train	Test
	ML		*M10*	
Bursty HDP-LDA	4904.2 ± 71.3	4992.9 ± 65.6	1959.4 ± 32.8	2265.2 ± 68.2
Non-parametric ATM	2238.2 ± 12.2	2460.3 ± 11.3	1562.9 ± 18.1	1814.0 ± 23.2
CNTM w/o network	1918.2 ± 4.3	2057.6 ± 3.6	912.7 ± 10.9	1186.1 ± 8.3
SCNTM ($\eta = 0$)	**1851.8*** ± 8.5	**1990.8*** ± 11.4	**824.0*** ± 12.0	**1048.3*** ± 21.4
	AvS		*CS*	
Bursty HDP-LDA	2460.4 ± 66.4	2612.8 ± 91.7	1509.2 ± 4.1	1577.8 ± 33.8
Non-parametric ATM	2199.7 ± 5.0	2481.7 ± 6.1	N/A	N/A
CNTM w/o network	1621.5 ± 19.5	2079.4 ± 2.6	1509.4 ± 4.1	1580.2 ± 32.6
SCNTM ($\eta = 0$)	**1620.6*** ± 2.2	**2028.0*** ± 10.9	**1275.3*** ± 14.0	**1530.8** ± 49.8
	Cora		*PubMed*	
Bursty HDP-LDA	678.1 ± 2.0	706.8 ± 17.0	**299.9** ± 0.2	**300.1** ± 1.2
CNTM w/o network	682.4 ± 1.5	702.5 ± 13.4	301.0 ± 0.2	301.2 ± 1.2
SCNTM ($\eta = 0$)	**621.1*** ± 6.7	**688.0** ± 15.7	312.3 ± 1.3	303.2 ± 1.2

Note that non-parametric ATM is not performed for the last three datasets due to the lack of authorship information in these datasets
Numbers that are bold denotes the best values
* signifies improvement at 5 % statistical significance over Bursty HDP-LDA and Non-parametric ATM

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

$$p(w_{dn} \mid \theta'_d, \phi) = \sum_k p(w_{dn} \mid z_{dn} = k, \phi_k)\, p(z_{dn} = k|\theta'_d) = \sum_k \phi_{kw_{dn}} \theta'_{dk}, \qquad (40)$$

again noting that the distributions ϕ' and θ are integrated out (see the method in "Appendix 5").

We can calculate the perplexity estimate for 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 4, showing the significantly (at 5 % significance level) better performance of SCNTM against the baselines on ML, M10 and AvS datasets. For these datasets, inclusion of citation information also provides additional improvement for model fitting, as shown in the comparison with CNTM without network component. For the CS, Cora and PubMed datasets, the non-parametric ATM was not performed due to the lack of authorship information. We note that the results for other η is not presented as they are significantly worse than $\eta = 0$. This is because the models are more restrictive, causing the likelihood to be worse. We like to point out that when no author is observed, the CNTM is more akin to a variant of HDP-LDA which uses PYP instead of DP, this explains why the perplexity results are very similar.

7.4 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 proportion in a topic distribution.

As mentioned in Sect. 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 are provided. We evaluate the clustering performance with *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. 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

$$\text{Purity}(\mathcal{S}, \mathcal{R}) = \frac{1}{D} \sum_k \max_j |r_k \cap s_j|, \quad \text{NMI}(\mathcal{S}, \mathcal{R}) = \frac{2\, I(\mathcal{S}; \mathcal{R})}{H(\mathcal{S}) + H(\mathcal{R})}, \tag{41}$$

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

$$I(\mathcal{S}; \mathcal{R}) = \sum_{k,\, j} \frac{|r_k \cap s_j|}{D} \log_2 \frac{D|r_k \cap s_j|}{|r_k||s_j|}, \quad H(\mathcal{R}) = -\sum_k \frac{|r_k|}{D} \log_2 \frac{|r_k|}{D}. \tag{42}$$

The clustering results are presented in Table 5. We can see that the SCNTM 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

Table 5 Comparison of clustering performance

Models	Purity	NMI	Purity	NMI
	M10		*AvS*	
Bursty HDP-LDA	0.66 ± 0.02	0.67 ± 0.01	$\mathbf{0.75} \pm 0.03$	$\mathbf{0.66} \pm 0.01$
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	$\mathbf{0.66} \pm 0.01$
SCNTM ($\eta = 0$)	0.67 ± 0.03	0.69 ± 0.02	0.72 ± 0.01	$\mathbf{0.66} \pm 0.00$
SCNTM ($\eta = 10$)	$\mathbf{0.73}^* \pm 0.02$	$\mathbf{0.72}^* \pm 0.01$	0.73 ± 0.01	$\mathbf{0.66} \pm 0.01$
SCNTM ($\eta = \infty$)	0.70 ± 0.03	0.70 ± 0.02	0.73 ± 0.02	$\mathbf{0.66} \pm 0.01$
	CS		*Cora*	
PMTLM	N/A	0.51	N/A	0.41
Bursty HDP-LDA	0.46 ± 0.11	0.63 ± 0.03	0.34 ± 0.03	0.58 ± 0.01
CNTM w/o network	0.51 ± 0.07	0.67 ± 0.02	0.37 ± 0.03	0.63 ± 0.01
SCNTM ($\eta = 0$)	0.51 ± 0.08	0.66 ± 0.02	0.39 ± 0.03	0.63 ± 0.02
SCNTM ($\eta = \infty$)	$\mathbf{0.54} \pm 0.10$	$\mathbf{0.69} \pm 0.04$	$\mathbf{0.47}^* \pm 0.06$	$\mathbf{0.66}^* \pm 0.03$
	PubMed			
PMTLM	N/A	0.27		
Bursty HDP-LDA	$\mathbf{0.53} \pm 0.04$	$\mathbf{0.73} \pm 0.01$		
CNTM w/o network	0.47 ± 0.04	0.69 ± 0.01		
SCNTM ($\eta = 0$)	0.46 ± 0.02	0.69 ± 0.01		
SCNTM ($\eta = \infty$)	0.52 ± 0.01	0.72 ± 0.01		

The best PMTML results are chosen for comparison, from Table 2 in Zhu et al. (2013)
Numbers that are bold denotes the best values
* signifies improvement at 5 % statistical significance over Bursty HDP-LDA and Non-parametric ATM

validation. 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 3).

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-topic network learned by the SCNTM.

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 SCNTM. The top words give us an idea of what the topics are about. In Table 6, we display some major topics extracted 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

Table 6 Topical summary for the ML, M10 and AvS datasets

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

The top words are extracted from the topic-word distributions ϕ learned by SCNTM

Table 7 Major authors and their main research area

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

Top words are extracted from the topic-word distribution ϕ_k corresponding to the dominant topic k of the author

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 focuses only on one particular topic, which is on Diabetes Mellitus.

8.2 Analysing authors' research area

In SCNTM, 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. In Table 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 markets. Again, we note that the topic labels are manually assigned to the authors based on the top words associated with their dominant topics.

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 extracted by SCNTM, using `Graphviz`.[11] On the ML, M10 and AvS datasets, we analyse the influential authors and their connections with the various topics learned by SCNTM. The influential authors are determined based on a measure we call author influence, which is the sum of the λ^- of all their publications, i.e. the influence of an author i is $\sum_d \lambda_d^- I(a_d = i)$. Note that a_d denotes the author of document d, and $I(\cdot)$ is the indicator function, as previously defined.

Figure 2 shows a snapshot of the author-topics network of the ML dataset. The pink rectangles in the snapshot represent the topics learned by SCNTM, 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

[11] http://www.graphviz.org/.

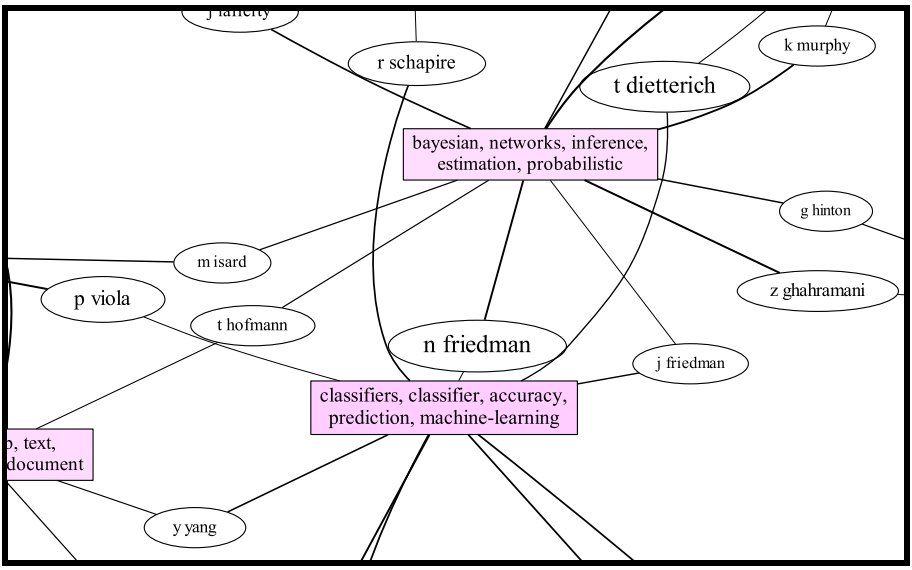

Fig. 2 Snapshot of the author-topics network from the ML dataset. The *pink rectangles* represent the learned topics, their intensity (*pinkness*) corresponds to the topic proportion. The *ellipses* represent the authors, their size corresponds to the author's influence in the corpus. The strength of the connections are given by the lines' thickness (Color figure online)

topic. Note that not all connections are shown, some of the weak connections are dropped to create a neater diagram. In Fig. 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[12] the full visualisation of the author-topics network learned from the CiteSeer^{X} datasets.

9 Diagnostics

In this section, we perform some diagnostic tests for the SCNTM. We assess the convergence of the MCMC algorithm associated with SCNTM 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 SCNTM.

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 Fig. 3, we show the time series plot of the training word log likelihood $\sum_{d,n} \log(p(w_{dn} \mid z_{dn}, \phi'))$ corresponds to the SCNTM trained with and without the network information. Recall that for SCNTM, the sampler for the topic

[12] https://drive.google.com/folderview?id=0B74l2KFRFZJmVXdmbkc3UlpUbzA (please download and view with a web browser for best quality).

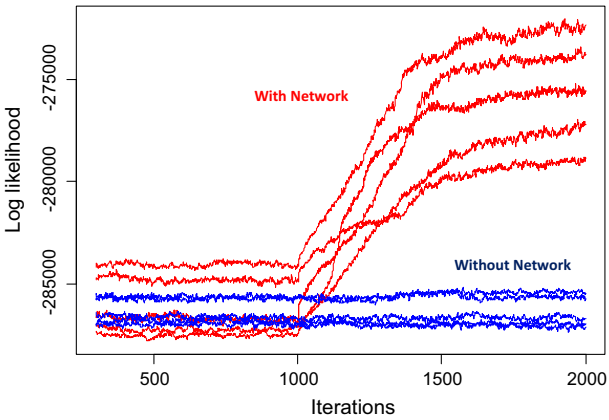

Fig. 3 (Coloured) Training word log likelihood versus iterations during training of the CNTM with and without the network component. The *red lines* show the log likelihoods of the SCNTM with the citation network while the *blue lines* represent the SCNTM without the citation network. The five runs are from five different folds of the Cora dataset (Color figure online)

model is first performed for 1,000 iterations before running the full MCMC algorithm. From Fig. 3, we can clearly see that the sampler converges quickly. For SCNTM, 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 about 95 %, which is very high, suggesting that the proposed MH algorithm is effective.

9.2 Inspecting document-topic hierarchy

As previously mentioned, modelling the 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.

9.3 Computation complexity

Finally, we briefly discuss the computational complexity of the proposed MCMC algorithm for the SCNTM. Although we did not particularly optimise our implementation for algorithm

Table 8 Time taken to perform 2,000 iterations of the MCMC algorithm given the statistics of the datasets

Datasets	Total words	Citations	Number of topics	Time (min)
ML	8,270,084	1,105,462	20	16,444
M10	595,918	77,222	50	1,845
AvS	1,102,608	54,601	30	2,092
CS	105,322	4,608	6	43
Cora	49,286	5,429	7	26
PubMed	1,332,869	44,335	3	397

The reported SCNTM run time corresponds to $\eta = \infty$

speed, the algorithm is of linear time with the number of words, the number of citations and the number of topics. All implementations are written in `Java`.

We implemented a general sampling framework that works with arbitrary PYP network, this allows us to test various PYP topic models with ease and without spending too much time in coding. 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 `hca`[13]). 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. The MCMC algorithm is constant time with respect to the number of authors.

Table 8 shows the average time taken to perform the MCMC algorithm for 2000 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.

10 Conclusions

In this paper, we have proposed the Supervised Citation Network Topic Model (SCNTM) as an extension of our previous work (Lim and Buntine 2014) to jointly model research publications and their citation network. The SCNTM makes use of the author information as well as the categorical labels associated with each document for supervised learning. The SCNTM performs text modelling with a hierarchical PYP topic model and models the citations with the Poisson distribution given the learned topic distributions. We also proposed a novel learning algorithm for the SCNTM, 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 sampler of a topic model. As discussed, our learning algorithm is intuitive and easy to implement.

The SCNTM 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

[13] http://mloss.org/software/view/527/.

and clustering evaluation. Additionally, incorporating supervision into the SCNTM 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 SCNTM, which allows us to have a quick look at the connection between the authors by way of their research areas.

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.

Appendix1: Delta method approximation

We employ the Delta Method to show that

$$\int q(\theta)\ \exp(-g(\theta))\,\mathrm{d}\theta \approx \exp(-g(\hat{\theta}))\int q(\theta)\,\mathrm{d}\theta \quad \text{for small } g(\hat{\theta}), \tag{43}$$

where $\hat{\theta}$ is the expected value according to a distribution proportional to $q(\theta)$, more specifically, define $p(\theta)$ as the probability density of θ, we have

$$\hat{\theta} = \mathbb{E}[\theta] = \int \theta\ p(\theta)\,\mathrm{d}\theta, \quad q(\theta) = \text{constant} \times p(\theta). \tag{44}$$

First we note that the Taylor expansion for a function $h(\theta) = \exp(-g(\theta))$ at $\hat{\theta}$ is

$$h(\theta) = \sum_{n=0}^{\infty} \frac{1}{n!}\left(h^{(n)}(\hat{\theta})\right)(\theta - \hat{\theta})^n, \tag{45}$$

where $h^{(n)}(\hat{\theta})$ denotes the n-th derivative of $h(\cdot)$ evaluated at $\hat{\theta}$:

$$h^{(n)}(\hat{\theta}) = \left(-g'(\hat{\theta})\right)^n\ h(\hat{\theta}). \tag{46}$$

Multiply Eq. 45 with $q(\theta)$ and integrating gives

$$\begin{aligned}\int q(\theta)\,h(\theta)\,\mathrm{d}\theta &= \sum_{n=0}^{\infty}\frac{1}{n!}\left(h^{(n)}(\hat{\theta})\right)\int q(\theta)\,(\theta-\hat{\theta})^n\,\mathrm{d}\theta \\ &= \sum_{n=0}^{\infty}\frac{1}{n!}\left(-g'(\hat{\theta})\right)^n\int q(\theta)\,(\theta-\hat{\theta})^n\,\mathrm{d}\theta.\end{aligned} \tag{47}$$

Since $g(\hat{\theta})$ is small, the term $\left(-g'(\hat{\theta})\right)^n$ becomes exponentially smaller as n increases. Here we let $\left(-g'(\hat{\theta})\right)^n \approx 0$ for $n \geq 2$. Hence, continuing from Eq. 47:

$$\begin{aligned}\int q(\theta)\,h(\theta)\,\mathrm{d}\theta \ &\approx\ h(\hat{\theta})\int q(\theta)\,\mathrm{d}\theta + \left(-g'(\hat{\theta})\right)h(\hat{\theta})\ \underbrace{\int q(\theta)\,(\theta-\hat{\theta})\,\mathrm{d}\theta}_{0} \\ &\approx\ h(\hat{\theta})\int q(\theta)\,\mathrm{d}\theta.\end{aligned} \tag{48}$$

Appendix 2: Keywords for querying the CiteSeer^{X} 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, pulverized coal, crude oil, fluidized 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.

Appendix 3: Recovering word counts from TF–IDF

The PubMed dataset (Sen et al. 2008) was preprocessed to TF–IDF (term frequency–inverse document frequency) format, i.e. the raw word count information is lost. Here, we describe how we recover the word count information, using a simple and reasonable assumption—that the least occurring words in a document only occur 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. TF–IDF is computed[14] as

$$t_{dw} = f_{dw} \times i_w, \quad f_{dw} = \frac{c_{dw}}{\sum_w c_{dw}}, \quad i_w = \log \frac{\sum_d 1}{\sum_d I(c_{dw} > 0)}, \tag{49}$$

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 Eq. 49. We can then determine the TF:

$$\begin{aligned} f_{dw} &= t_{dw}/i_w \\ &= t_{dw} \times \left(\log \frac{\sum_d 1}{\sum_d I(t_{dw} > 0)} \right)^{-1}. \end{aligned} \tag{50}$$

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}. Fortunately, since we are working with natural language, it is reasonable to assume that the least occurring words in a document only occur once, or mathematically,

$$c_{dw} = 1 \quad \text{for} \quad w = \arg\min_w f_{dw}. \tag{51}$$

[14] 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.

Thus we can work out the normaliser $\sum_w c_{dw}$ and recover the word counts for all words in all documents.

$$\sum_w c_{dw} = \frac{1}{\min_w f_{dw}}, \quad c_{dw} = f_{dw} \times \sum_w c_{dw}. \tag{52}$$

Appendix 4: Exclusion words to detect invalid authors

Below is a list of words we use to filter out invalid authors during 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.

Appendix 5: Integrating out probability distributions

Here, we show how to integrate out probability distributions using the expectation of a PYP:

$$\begin{aligned} p(w_{dn}|z_{dn}=k,\phi_k) &= \int_{\phi'_{dk}} p(w_{dn},\phi'_{dk}|z_{dn},\phi_k) \\ &= \int_{\phi'_{dk}} p(w_{dn}|z_{dn},\phi'_{dk})\, p(\phi'_{dk}|\phi_k) \\ &= \int_{\phi'_{dk}} \phi'_{dkw_{dn}}\, p(\phi'_{dk}|\phi_k) \\ &= \mathbb{E}[\phi'_{dkw_{dn}}|\phi_k] \\ &= \phi_{kw_{dn}}, \end{aligned} \tag{53}$$

where $\mathbb{E}[\cdot]$ denotes the expectation value. We note that the last step (Eq. 53) 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). A similar approach can be taken to integrate out the θ in Eq. 40.

References

Blei, D., Ng, A., & Jordan, M. (2003). Latent Dirichlet Allocation. *JMLR*, *3*, 993–1022.

Buntine, W., & Hutter, M. (2012). A Bayesian view of the Poisson-Dirichlet process. ArXiv e-prints 1007.0296v2.

Buntine, W., & Mishra, S. (2014). Experiments with non-parametric topic models. In *KDD* (pp 881–890). ACM.

Carpenter, B. (2004). Phrasal queries with LingPipe and Lucene: Ad hoc genomics text retrieval. In *TREC*.

Casella, G., & Robert, C. P. (1996). Rao-Blackwellisation of sampling schemes. *Biometrika*, *83*(1), 81–94.

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

Chen, C., Du, L., & Buntine, W. (2011). Sampling table configurations for the hierarchical Poisson-Dirichlet process. In *ECML* (pp. 296–311). Springer.

Goldwater, S., Griffiths, T., & Johnson, M. (2011). Producing power-law distributions and damping word frequencies with two-stage language models. *JMLR*, *12*, 2335–2382.

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

Han, H., Zha, H., & Giles, C. L. (2005). Name disambiguation in author citations using a K-way spectral clustering method. In *JCDL* (pp. 334–343). ACM.
Kataria, S., Mitra, P., Caragea, C., & Giles, C. L. (2011). Context sensitive topic models for author influence in document networks. In *IJCAI* (pp. 2274–2280). AAAI Press.
Lim, K. W., & Buntine, W. (2014). Bibliographic analysis with the citation network topic model. In *ACML* (pp. 142–158).
Lim, K. W., Chen, C., & Buntine, W. (2013). Twitter-network topic model: A full Bayesian treatment for social network and text modeling. In *NIPS Topic Model workshop*.
Liu, L., Tang, J., Han, J., Jiang, M., & Yang, S. (2010). Mining topic-level influence in heterogeneous networks. In *CIKM* (pp. 199–208). ACM.
Liu, Y., Niculescu-Mizil, A., & Gryc, W. (2009). Topic-link LDA: Joint models of topic and author community. In *ICML* (pp. 665–672). ACM.
Lui, M., & Baldwin, T. (2012). langid.py: An off-the-shelf language identification tool. In *ACL* (pp. 25–30). ACL.
Manning, C., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge: Cambridge University Press.
McCallum, A. K. (2002). MALLET: A machine learning for language toolkit. http://www.cs.umass.edu/~mccallum/mallet.
Mimno, D., McCallum, A. (2007). Mining a digital library for influential authors. In *JCDL* (pp. 105–106). ACM.
Nallapati, R., Ahmed, A., Xing, E., & Cohen, W. (2008). Joint latent topic models for text and citations. In *KDD* (pp. 542–550). ACM.
Oehlert, G. W. (1992). A note on the delta method. *The American Statistician*, *46*(1), 27–29.
Pitman, J. (1996). Some developments of the Blackwell–Macqueen urn scheme. Lecture Notes—Monograph Series (pp. 245–267).
Rosen-Zvi, M., Griffiths, T., Steyvers, M., & Smyth, P. (2004). The author-topic model for authors and documents. In *UAI* (pp. 487–494). AUAI Press.
Sato, I., & Nakagawa, H. (2010). Topic models with power-law using Pitman–Yor process. In *KDD* (pp. 673–682). ACM.
Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., & Eliassi-Rad, T. (2008). Collective classification in network data. *AI Magazine*, *29*(3), 93–106.
Tang, J., Sun, J., Wang, C., & Yang, Z. (2009). Social influence analysis in large-scale networks. In *KDD* (pp. 807–816). ACM.
Teh, Y. W. (2006a). A Bayesian interpretation of interpolated Kneser–Ney. Tech. rep., School of Computing, National University of Singapore.
Teh, Y. W. (2006b). A hierarchical Bayesian language model based on Pitman–Yor processes. In *ACL* (pp 985–992). ACL.
Teh, Y. W., Jordan, M. (2010). Hierarchical Bayesian nonparametric models with applications. In N. L. Hjort, C. Holmes, P. Müller, & S. G. Walker (Eds.), *Bayesian nonparametrics: Principles and practice* (Chap. 5). Cambridge University Press.
Tu, Y., Johri, N., Roth, D., & Hockenmaier, J. (2010). Citation author topic model in expert search. In *COLING* (pp. 1265–1273). ACL.
Wallach, H., Mimno, D., & McCallum, A. (2009). Rethinking LDA: Why priors matter. In *NIPS* (pp. 1973–1981).
Weng, J., Lim, E. P., Jiang, J., & He, Q. (2010). TwitterRank: Finding topic-sensitive influential Twitterers. In *WSDM* (pp. 261–270). ACM.
Zhu, Y., Yan, X., Getoor, L., & Moore, C. (2013). Scalable text and link analysis with mixed-topic link models. In *KDD* (pp 473–481). ACM.