Experiments with Non-parametric Topic Models

Wray Buntine[*]
Monash University
Clayton, VIC, Australia
wray.buntine@monash.edu

Swapnil Mishra
RSISE, The Australian National University
Canberra, ACT, Australia
swapnil.mishra@anu.edu.au

ABSTRACT

In topic modelling, various alternative priors have been developed, for instance asymmetric and symmetric priors for the document-topic and topic-word matrices respectively, the hierarchical Dirichlet process prior for the document-topic matrix and the hierarchical Pitman-Yor process prior for the topic-word matrix. For information retrieval, language models exhibiting word burstiness are important. Indeed, this burstiness effect has been show to help topic models as well, and this requires additional word probability vectors for each document. Here we show how to combine these ideas to develop high-performing non-parametric topic models exhibiting burstiness based on standard Gibbs sampling. Experiments are done to explore the behavior of the models under different conditions and to compare the algorithms with previously published. The full non-parametric topic models with burstiness are only a small factor slower than standard Gibbs sampling for LDA and require double the memory, making them very competitive. We look at the comparative behaviour of different models and present some experimental insights.

Categories and Subject Descriptors

I.7 [**Document and Text Processing**]: Miscellaneous; I.2.6 [**Artificial Intelligence**]: Learning

Keywords

topic modelling; experimental results; non-parametric prior; text

1. INTRODUCTION

Topic models are now a recognised genre in the suite of exploratory software available for text data mining and other semi-structured tasks. Moreover, it is also recognised that a broad class of variants can be developed based around extended graphical models that capture some additional domain requirements. Examples of these extensions include author-topic modelling [19], document segmentation [8] and word-sense disambiguation [1], however the list is extensive. Here we consider the task of improving the vanilla topic model, but in the back of our mind is the requirement that the techniques need to easily be transferred to the host of variants that make up a significant part of use in applications.

While the standard model is Latent Dirichlet Allocation (LDA), and many techniques exist to scale this up significantly, better quality topic models are available. Two clear and related innovations are available here. The first is the use of more sophisticated priors for the probability vectors rather than the simple symmetric Dirichlet [25], and the second is the use of non-parametric methods, best characterised by the HDP-LDA model [24], again to improve the priors, the standard being the hierarchical Dirichlet process. These have the goals of better estimating prior topic or word proportions, and also estimating the "right" number of topics. Note these are related in that a technique to estimate the proportions for different topics can also make some topics insignificant, thus effectively changing the number of topics.

Another innovation, not so well known, is to model *burstiness* [5], which is the idea that once we see a word, we can expect to see it again. Consider the following news snippet:

> Despite their separation, Charles and Diana stayed close to their *boys William* and Harry. Here, they accompany the *boys* for 13-year-old *William*'s first day school at Eton College on Sept. 6, 1995, with housemaster Dr. Andrew Gayley looking on.

We see here that two words, "boys" and "William" appear twice. In the information retrieval literature, a related phenomenen is the notion of *eliteness* [12] whereby words are said to have different "levels" of occurrence, and this influenced the development of the dominant relevance paradigm in information retrieval [18].

These innovations, non-parametric priors and burstiness, have so far been bogged down by computationally intensive techniques that prevent their wider use. The original implementation of DCMLDA for burstiness [5] was only able to be applied to small data sets of less than a 1000 documents. A theme of research in recent machine learning conferences have been a variety of alternative algorithms for inference with HDP-LDA [22, 27, 2]. One technique of note is the use of stochastic variational methods that allows application to streaming data [27, 13]. An excellent theoretical and em-

[*]Part of this author's contribution was done while at NICTA, Canberra.

KDD'14, August 24–27, 2014, New York, NY, USA.
ACM 978-1-4503-2956-9/14/08.
http://dx.doi.org/10.1145/2623330.2623691.

pirical comparison of a variety of sampling and variational methods can be found in [20].

These newer algorithms, however, are usually based on the standard stick-breaking formulation for Dirichlet processes and variational methods for these and the simpler Dirichlet distribution. Recently, an alternative sampling scheme for the more general Pitman-Yor process has been developed [4] called *table indicator sampling*. It uses prebuilt tables of second order generalised Stirling numbers, and the scheme has seen use on problems such as document segmentation [9] and topic models on structured text [7]. Its key advantages are that it requires no dynamic memory for implementation, and that the convergence is usually significantly faster and better quality (it is a collapsed Gibbs sampler) [6].

This table indicator sampling allows easy development of non-parametric topic models including HDP-LDA, its extension to the Pitman-Yor process (PYP) and power-law models [21] which place the PYP on the topic-word matrix. What is interesting, however, which is our first major contribution, is that we can easily extend this broader variety of non-parametric topic models by adding a burstiness component on the front end of the model. Moreover, an implementation trick lets this be done with little additional memory/time overhead. These models are available in our recently MLOSS-released open source multi-core software `hca`[1].

The resulting non-parametric LDA algorithms turn out to be fast. Generally, they are a factor or two (in memory and time) slower than standard Gibbs implementations of vanilla LDA and typically 5 times faster than comparable variational HDP-LDA implementations but with the same memory requirements. Although, we also show that Mallet's [16] asymmetric-symmetric LDA is a form of truncated HDP-LDA and it is an order of magnitude faster again. The experimental results show improvement over all the existing methods in perplexity, including Mallet. We also conduct a number of experiments to explore the nature of the new algorithms. The full experiments with burstiness are the first done at moderate scale with these sort of models and the experimental insights are our second major contribution. Our implementation runs bursty HDP-LDA with K=1000 topics on 800k news articles at 10 minutes per major Gibbs cycle using a standard 8-core CPU.

We first discuss, at a general level, how we use Pitman-Yor processes and the nature of the inference with them, in Section 2. Section 3 presents the different models used here. Because our inference schemes are standard block Gibbs samplers in the table indicator framework, we do not detail the algorithms here other than describing how we implement burstiness. Section 4 then presents our experimental setup, and a sequence of empirical investigations follow in Section 5.

2. THE HIERARCHICAL PITMAN-YOR PROCESS

Here we briefly review the methods used for inference on the hierarchical Pitman-Yor processes that one can see embedded in the model we use [3]. Those not needing to understand the fundamentals of the sampling methods can skip this section.

All our samplers use standard block table indicator Gibbs samplers for the network of Pitman-Yor processes [4, 9] and adaptive rejection sampling [10] for the many hyperparameters. Slice sampling is usually of similar performance but can suffer with the extremely peaked posteriors for the concentration parameter of a Pitman-Yor process.

We use the Pitman-Yor process as a distribution on a probability vector. The distribution has a mean (*i.e.*, another probability vector), a variance parameter represented as a concentration (usually given as b_X when on vector $\vec{X}$), and a third parameter called discount (usually given as a_X when on vector $\vec{X}$), so one has $\vec{p} \sim \text{PYP}\left(a_p, b_p, \vec{\theta}\right)$. The Pitman-Yor process, when used in this way, can be used hierarchically to form distributions on a network of probability vectors.

The inference on a network of probability vectors is based on a basic property of species sampling schemes [14] that is best understood using the framework of message passing over networks. Figure 1a shows the context of a probability vector $\vec{p}$ having a Pitman-Yor process with base distribution $\vec{\theta}$. Two multinomial style likelihood messages are passed up to $\vec{p}$ with counts $\vec{n}$ and $\vec{m}$. The standard message then passed on from $\vec{p}$ to $\vec{\theta}$ using the multinomial-Dirichlet distribution [3] is a complex set of gamma functions obtained using the normalising term for a Dirichlet distribution, represented in the figure as $l_p^{Dir}(\vec{\theta})$ (assuming a concentration at the node of b_p)

$$l_p^{Dir}(\vec{\theta}) \ = \ \frac{\Gamma(b_p)}{\Gamma(b_p+N+M)} \prod_k \frac{\Gamma\left(b_p\theta_k+n_k+m_k\right)}{\Gamma(b_p\theta_k)} \ , \quad (1)$$

where the total statistics are $N=\sum_k n_k$ and $M=\sum_k m_k$. This functional complexity on $\vec{\theta}$ prevents any further network inference.

Figure 1b shows the alternative after marginalising out the vector $\vec{p}$ using Pitman-Yor process theory [3]. One however must introduce a new latent count vector $\vec{t}$ that represents the fraction of the data $\overrightarrow{n+m}$ that passes up in a message to $\vec{\theta}$. These are called *table multiplicities* and they correspond to the number of tables in the corresponding Chinese restaurant process (CRP) [24]. The multiplicites have a bounding constraint $t_k \le n_k+m_k$ and moreover $t_k \equiv 0$ if and only if $n_k+m_k \equiv 0$. Thus at the expense of introducing a latent count vector ($\vec{t}$) one gets a simple multinomial likelihood passed up the hierarchy, albiet with a complex looking but O(1) normalising constant, in the form

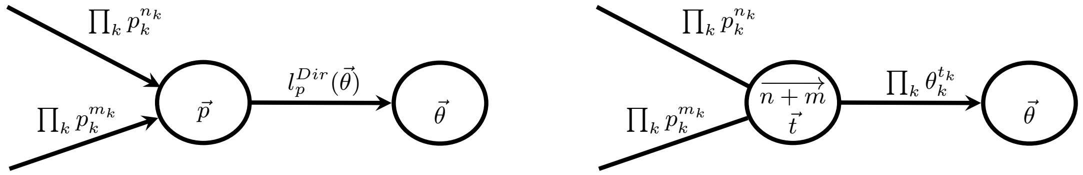
$$l_p^{PYP}(\vec{\theta}) \ = \ \frac{(b_p|a_p)_T\Gamma(b_p)}{\Gamma(b_p+N+M)} \prod_k S_{t_k,a_p}^{n_k+m_k}\theta_k^{t_k} \ , \quad (2)$$

where the total $T=\sum_k t_k$ and $(x|y)_T$ denotes the Pochhammer symbol, $(x|y)_T = x(x+y)\ldots(x+(T-1)y)$. $S_{t,a}^c$ is a generalized second-order Stirling number [23]. Libraries[2] exist for efficiently dealing with this making it in most cases an $O(1)$ computation.

The counts of $\vec{t}$ then contribute to the data at the parent node $\vec{\theta}$ and thus its posterior probability. Thus network inference is feasible, and moreover no dynamic memory is required, unlike CRP methods, because $\vec{t}$ is the same dimension as $\overrightarrow{n+m}$. Sampling the $\vec{t}$ directly, however, leads

[1] See `http://mloss.org/software/view/527/`

[2] See `https://mloss.org/software/view/424/` and `https://mloss.org/software/view/528`



(a) Embedded probability vector with messages. (b) Embedded count vectors after marginalising.

Figure 1: Computation with species sampling models.

to a poor algorithm because they may have a large range (above, $0 \leq t_k \leq n_k + m_k$) and the impact of data at the node $\vec{p}$ on the node $\vec{\theta}$ is buffered by $\vec{t}$, leading to poor mixing.

Table indicators are introduced by [4] to allow the sampling of the table multiplicity vectors $\vec{t}$ to be done incrementally and thus allow more rapid mixing and simpler sampling. Note the table indicators are Boolean values indicating if the current data item increments the table multiplicity at its node and thus the data item contributes to the message to the parent. The assignment of indicators to data can be done because of the above constraint $t_k \leq n_k + m_k$. So the data item contributes a $+1$ to $n_k + m_k$ and the matched table indicator contributes either 0 or $+1$ to t_k, which is the change in the message to the parent. If there is a grandparent node, then a corresponding table indicator in the parent node might also propagate a $+1$ up to the grandparent.

For inference on a network of such vectors, each probability vector node contributes a factor to the posterior probability. For the above example with table indicators this is given by Formula (3) [3]

$$\frac{(b_p|a_p)_T \Gamma(b_p)}{\Gamma(b_p + N + M)} \prod_k S^{n_k+m_k}_{t_k,a_p} \binom{n_k + m_k}{t_k}^{-1} , \qquad (3)$$

where the addition of the $\binom{n_k+m_k}{t_k}$ term over Equation (2) simply divides by the number of choices there are for picking the t_k boolean table indicators to be on out of a possible $n_k + m_k$.

In sampling, a data point coming from the node source for $\vec{n}$ contributes a $+1$ to n_k (for some k), and either contributes a $+1$ or a 0 to t_k depending on the value of the table indicator. If $n_k = t_k = 0$ initially, then it must contribute a $+1$ to t_k, so there is no choice. The change in posterior probability of Formula (3) due to the new data point at this node is, given the Boolean indicator r_l

$$\Big((t_k + 1)\,(b_p + T * a_p)\, S^{n_k+m_k+1}_{t_k+1,a_p} \Big)^{r_l \equiv 1}$$
$$\Big((n_k + m_k - t_k + 1) S^{n_k+m_k+1}_{t_k,a_p} \Big)^{r_l \equiv 0}$$
$$\Big((n_k + m_k + 1)\,(b_p + N + M)\, S^{n_k+m_k}_{t_k,a_p} \Big)^{-1} \qquad (4)$$

depending on the value of the table indicator r_l for the data point. In a network, one has to jointly sample table indicators for all reachable ancestor nodes in the network, and standard discrete graphical model inference is done in closed form. Examples are given by [7, 8].

For estimation, one requires the expected probabilities $\mathbb{E}_{\vec{n},\vec{m},\vec{t},\vec{\theta}}[\vec{p}]$ at a node. In the table indicator framework this is harder to compute. Fortunately, with a trivial change of latent variables (drop the table indicators and reintroduce the table occupancies for the CRP) we get the usual estimation formula for the CRP [23] given by Equation (5)

$$\mathbb{E}_{\vec{n},\vec{m},\vec{t},\vec{\theta}}[\vec{p}] \;=\; \frac{b_p + T * a_p}{b_p + N + M} \vec{\theta} + \frac{\vec{n} + \vec{m} - a_p \vec{t}}{b_p + N + M} \,. \qquad (5)$$

Since this does not involve knowing the table occupancies for the CRP, no additional sampling is needed to compute the formula, just the existing counts (*i.e.*, $\vec{n}, \vec{m}, \vec{t}$) are used. Moreover, we know the estimates normalise correctly.

3. MODELS

3.1 Basic Models

The basic non-parametric topic model we consider is given in Figure 2. Here, the document-topic proportions $\vec{\theta}_i$ (for i running over documents) have a PYP with mean $\vec{\alpha}$ and the topic-word proportions $\vec{\phi}_k$ (for k running over topics) have a PYP with mean $\vec{\beta}$. The mean vectors $\vec{\alpha}$ and $\vec{\beta}$ correspond to the asymmetric priors of [25].

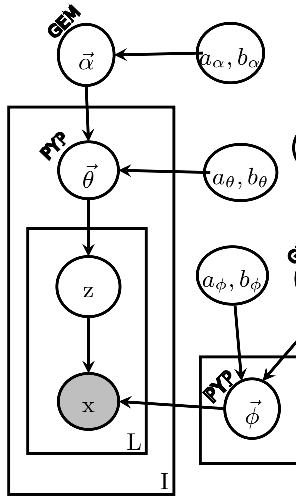


Figure 2: Non-parametric topic model.

While we show $\vec{\alpha}$ and $\vec{\beta}$ having a GEM prior [15, 24] in the figure, allowing different priors covers a range of LDA styles, as shown in Table 1. For instance, when $\vec{\alpha}$ is finite and the discount for the PYP on $\vec{\theta}$, a_θ, is zero, then $\vec{\theta} \sim$ Dirichlet$(b_\theta \vec{\alpha})$. Thus the two PYPs in the figure can be configured to be Dirichlets, giving the standard LDA set-up for $\vec{\theta}$ and likewise for $\vec{\phi}_k$. The GEM is equivalent to the stick-breaking prior that is at the core of a DP or PYP, so

Table 1: Family of LDA Models. The "tr" abbreviates "truncated" and "symm" abbreviates "symmetric".

	$\vec{\alpha}$ prior	a_α	a_θ	$\vec{\beta}$ prior	a_ϕ
LDA	finite $\vec{u}$	0	0	finite $\vec{u}$	0
tr. HDP-LDA	tr. GEM	0	0	finite $\vec{u}$	0
tr. HDP-LDA	symm. Dir.	0	0	finite $\vec{u}$	0
tr. NP-LDA	tr. GEM	-	0	tr. GEM	-

using this with $\vec{\alpha}$ and a truncated K, and setting $\vec{\phi}_k$ up to be Dirichlet distributed as just shown, we have truncated HDP-LDA. Notice there are different ways of provided a truncated prior to ensure a fixed dimensional $\vec{\alpha}$. The truncated GEM is used in various versions of truncated HDP-LDA [22, 27], and the simpler truncation, just using a Dirichlet, is implicit in the asymmetric priors of [25]. That is, the asymmetric-symmetric (AS) variant of LDA [25] is equivalent to a truncated HDP-LDA. This means that `Mallet` [16] has implemented a truncated HDP-LDA (via AS-LDA) since 2008, and it is indeed both one of the fastest and the best performing.

Thus we reproduce several alternative variants of LDA [25], as well as truncated versions of HDP-LDA, HPYP-LDA and a fully non-parametric asymmetric version (with the truncated GEM prior on both $\vec{\alpha}$ and $\vec{\beta}$) we refer to as NP-LDA. Sampling algorithms for dealing with the HPYP-LDA case are from earlier work [4], and the other cases are similar.

3.2 Bursty Models

The extension with burstiness we consider [5] is given in Figure 3. Here, each topic $\vec{\phi}_k$ is specialised to a variant specific to each document i, $\vec{\psi}_{k,i}$. Thus

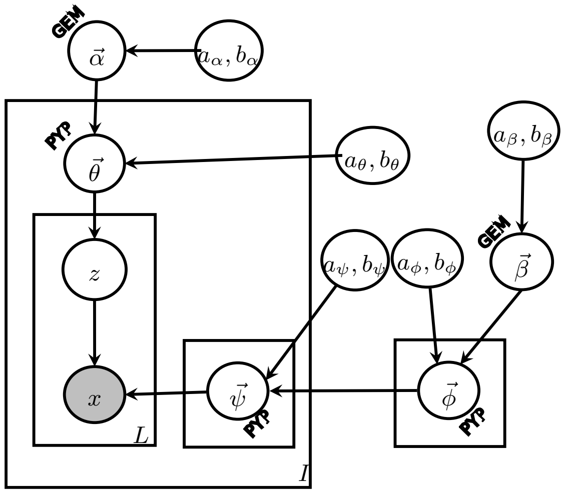


Figure 3: Model with topic burstiness.

$$\vec{\psi}_{k,i} \sim \mathrm{PY}\left(a_\phi, b_{\phi,k}, \vec{\phi}_k\right) \ .$$

On the surface one would think introducing potentially $K * W$ (number of words by number of topics) new parameters *for each document*, for the $\vec{\psi}_k$, seems statistically impractical. In practice, the $\vec{\psi}_k$ are marginalised out during inference and book-keeping only requires a small number of additional latent variables. Note that each topic k has its own concentration parameter $b_{\phi,k}$. This feature will be illustrated in Subsection 5.5.

3.3 Inference with Burstiness

In LDA style topic modelling using our approach, we get a formula for sampling a new topic z for a word w in position l in a document d. Suppose all the other data and the rest of the document is $\mathcal{D}_{-(d,l)}$ and this is some model $\mathcal{M}$ (maybe NP-LDA or LDA, *etc.*) with hyperparameters. Then denote this Gibbs sampling formula as $p\left(z \mid w, \mathcal{D}_{-(d,l)}, \mathcal{M}\right)$. For LDA, this is just the standard collapsed Gibbs sampling formula [11]. It also forms the first step of the block Gibbs sampler we use for HDP-LDA [4]: first we sample the topic z, and then we sample the various table indicators give z in the model.

The burstiness model built on $\mathcal{M}$, denote it $\mathcal{M}$-$\mathcal{B}$, is sampled using $p\left(z \mid w, \mathcal{D}_{-(d,l)}, \mathcal{M}\text{-}\mathcal{B}\right)$ which is computed using $p\left(z \mid w, \mathcal{D}_{-(d,l)}, \mathcal{M}\right)$. Thus we say the burstiness model $\mathcal{M}$-$\mathcal{B}$ is a front end to the Gibbs sampler. At the position l in a document we have a word of type w and wish to resample its topic $z = k$. Let $n_{w,k}$ be the number of other existing words of the type w already in topic k for the current document, and let $s_{w,k}$ be the corresponding table multiplicities. They are statistics for the parameters $\vec{\psi}_k$ in the burstiness model. Note by keeping track of which words in a document are unique, one knows that $n_{w,k} = 0$ for those words, thus computation can be substantially simplified. Let $N_{\cdot,k}$ and $S_{\cdot k}$ be the corresponding totals for the topic k in the document (*i.e.*, summed over words). The matrices of counts $n_{w,k}$ and $s_{w,k}$ and vectors $N_{\cdot,k}$ and $S_{\cdot k}$ can be recomputed as each document in processed in time proportional to the length of the document.

The Gibbs sampling probability for choosing $z = k$ at position l for the burstiness model is obtained using Equation (4).

$$\begin{aligned} & p\left(z=k \mid w, \mathcal{D}_{-(d,l)}, \mathcal{M}\text{-}\mathcal{B}\right) \ \propto \qquad (6) \\ & p\left(z \mid w, \mathcal{D}_{-(d,l)}, \mathcal{M}\right) \frac{b_{\psi,k}+a_\psi S_{\cdot k}}{b_{\psi,k}+N_{\cdot,k}} \frac{s_{w,k}+1}{n_{w,k}+1} \frac{S^{n_{w,k}+1}_{s_{w,k}+1,a_\psi}}{S^{n_{w,k}}_{s_{w,k},a_\psi}} \\ & \quad + \quad \frac{1}{b_{\psi,k}+N_{\cdot,k}} \frac{n_{w,k}-s_{w,k}+1}{n_{w,k}+1} \frac{S^{n_{w,k}+1}_{s_{w,k},a_\psi}}{S^{n_{w,k}}_{s_{w,k},a_\psi}} \ . \end{aligned}$$

This has a special case when $s_{w,k} = n_{w,k} = 0$ of

$$p\left(z \mid w, \mathcal{D}_{-(d,l)}, \mathcal{M}\right) \frac{b_{\psi,k}+a_\psi S_{\cdot k}}{b_{\psi,k}+N_{\cdot,k}} \ . \qquad (7)$$

Once topic $z = k$ is sampled, the second term of Equation (6) is proportional to the probability that the table indicator for word w in the $\vec{\psi}_k$ PYP is zero, it does not contribute data to the parent node $\vec{\phi}_k$, *i.e.*, the original model $\mathcal{M}$ will ignore this data point. The first term of Equation (6) is proportional to the probability that the table indicator is one, so it does contribute data to the parent node $\vec{\phi}_k$, *i.e.*, back to the original model $\mathcal{M}$. This table indicator is sampled according to the two terms and the $n_{w,k}, s_{w,k}, N_{\cdot,k}, S_{\cdot k}$ are all updated. If the table indicator is one then the original model $\mathcal{M}$ processes the data point in the manner it usually would.

Thus Equation (6) is used to filter words, so we refer to it as the burstiness front-end. Only words with table indicators of one are allowed to pass through to the regular model $\mathcal{M}$ and contribute to its statistics for $\vec{\phi}_k$ and, for instance, any further PYP vectors in the model.

4. EXPERIMENTAL SETUP

4.1 Implementation

The publically available `hca` suite used in these experiments is coded in C using 16 and 32 bit integers where needed for saving space. All data preparation is done using the DCA-Bags[3] package, a set of scripts, and input data can be handled in a number of formats including the LdaC format. All algorithms are run on a desktop with an Intel(R) Core(TM) i7 8-core CPU (3.4Ghz) using a single core.

The algorithms have no dynamic memory, so we set the maximum number of topics K ahead of time. This is like the truncation level in variational implementations of HDP-LDA. Moreover, initialisation is done by setting the number of topics to this maximum and randomly assigning words to topics. Other authors [22] report initialising to the maximum number of topics, rather than 1, leads to substantially better results, an experimental finding with which we agree.

Note, inference and learning for burstiness requires the word by topic counts $n_{w,k}$ and word by topic multiplicities $s_{w,k}$ be maintained for each document, as well as their totals. There is an implementation trick used to achieve space efficiency here. First one computes, for each document, which words appear more than once in the document (*i.e.*, those for which $n_{w,k}$ can become greater than 1). These words require special handling, the full Equation (6), and lists of these are stored in preset variable length arrays. Words that occur only once in a document are easy to deal with since their sampling is governed by Equation (7) and no sampling of the table indicator is needed. Second, the count and multiplicity statistics (the $n_{w,k}$ and $s_{w,k}$ which are statistics for $\vec{\psi}_{i,k}$) are not stored but recomputed as each document is about to be processed. Moreover, this only needs to be done for words appearing more than once in the document (hence why lists of these are prestored). All one needs to recompute these statistics is the Boolean table indicators and the topic assignments. The statistics $n_{w,k}, s_{w,k}$ can be recomputed in time proportional to the length of the document.

4.2 Data

We have used several datasets for our experiments, the PN, MLT, RML, TNG, NIPS and LAT datasets. Not all data sets were used in all comparisons.

The PN dataset is taken from 805K News articles (Reuters RCV1) using the query "person", excluding stop words and words appearing <5 times. The MLT dataset is abstracts from the JMLR volumes 1-11, the ICML years 2007-2011, and IEEE Trans.of PAMI 2006-2011. Stop words were discarded along with words appearing <5 or >2900 times. The RML dataset is the Reuters-21578 collection, made up using standard ModLewis split. The TNG dataset is the 20-newsgroup dataset using the standard split. For both stop words were discarded along with words appearing <5 times. The LAT dataset is the LA Times articles from TREC disk 4. Stop words were discarded along with words appearing <10 times. Only words made up entirely of alphabetic characters or dashes were allowed. Roweis' NIPS dataset[4] was left as is.

Table 2: Characteristic Sizes of Datasets

	PN	MLT	RML	TNG	LAT	NIPS
W	26037	4662	16994	35287	78953	13649
D	8616	2691	19813	18846	131896	1740
T	1000	306	6188	7532	0	348
N	1.76M	224k	1.27M	1.87M	34.5M	23.0M

Characteristics of these six datasets are given in Table 2, where dictionary size is W, number of documents (including test) is D, number of test documents is T and total number of words in the collection is N.

4.3 Evaluation

The algorithms are evaluated on two different measures, test sample perplexity and point-wise mutual information (PMI). Perplexity is calculated over test data and is done using document completion [26], known to be unbiased and easy to implement for a broad class of models. The document completion estimate is averaged over 40 cycles per document done at the end of the training run and uses a 80-20% split, so every fifth word is used to evaluate perplexity and the remaining to estimate latent variables for the document. Topic comprehensibility can be measured in terms of PMI [17]. It is done by measuring average word association between all pairs of words in the top-10 topic words (using the English Wikipedia articles). Here the PMI reported is average across all topics. PMI files are prepared with the `DCA-Bags` package using `linkCoco` and projected onto the specific data-sets using `cooc2pmi.pl` in the `hca` suite.

We also compare results with two other systems, `online-hdp` [27] is a stochastic variational algorithm for HDP-LDA coded in Python from C. Wang[5], and `HDP` a Matlab+C combination doing Gibbs sampling from Y.W. Teh. To do the comparisons, at various timepoints we take a snapshot of the $\vec{\alpha}$ vector and the $\vec{\phi}_k$ vectors. This is already supported in `onlinehdp`, and C. Chen provided the support for this task with `HDP`. We then load these values along with the hyperparameter settings into `hca` and use its document completion and PMI evaluation options "-V -p -hdoc,5." In this way, all algorithms are compared using identical software.

5. EXPERIMENTS

5.1 Runtime Comparisons

To see how the algorithms work at scale, we consider the cycle times and memory requirements of the different versions running on the full LAT data set. These are given in Table 3. Cycle times in minutes are for a full pass through all documents and memory requirements are given in megabytes. LDA, HDP-LDA (where $a_\theta = 0$) and NP-LDA are as described in Section 3. The right half of the table gives performance for the burstiness model of Figure 3. Note only a portion of the computation is linear in K so, for instance NP-LDA with burstiness using $K = 2000$ topics on the same dataset takes roughly 90 minutes a cycle and 2.43GB memory. Moreover, given it is coded in Python using inefficient

[3] http://mloss.org/software/view/522/

[4] `http://www.cs.nyu.edu/~roweis/data.html`

[5] Some C++ versions also exist.

Table 3: Cycle times and memory requirements on the LA Times TREC 4 data using $K = 500$ topics. "Burst" is the burstiness version.

	w/out Burst		with Burst	
Alg.	mins.	Mb	mins.	Mb
LDA	11	630	20	690
HDP-LDA	20	760	30	850
NP-LDA	35	840	45	930
`onlinehdp`	236	1800		

allocation, `onlinehdp` has comparable memory requirements to `hca`. In subsequent experiments, we also saw `HDP` required 5-7 times more memory than `hca`.

Experiments show that the convergence rates (in cycle counts not time) are similar for the various Gibbs algorithms (LDA, Burst LDA, Burst NP-LDA, etc.). Gibbs for full non-parametric LDA with the burstiness front end gives substantial improvements over vanilla Gibbs LDA while requiring only 50% more memory and 3 times greater computation time. Note that the table indicator samplers have previously been reported to give 1-2% improvement in perplexity over Chinese restaurant samplers [4], which in turn retain a substantial improvement over earlier variational algorithms for HDP-LDA [22].

We find that sampling hyper-parameters (for instance, discounts and concentrations of Pitman-Yor processes) to be important for performance. A substantial part of the time for topic burstiness is hyper-parameter sampling, something that is usually less than 5% for the other models. This is because the model has a different concentration parameter for every topic, thus much more of the inefficient adaptive rejection sampling is done for the bursty models versus others.

5.2 General Results

A subset of the results are presented in Table 4 and some informative plots given in Figure 4 and Figure 5. These represent the average values computed over 4 independent runs. Note that the differences between `hca`'s "Burst HDP" and "Burst NP-LDA" in the table are not significant at the 5% level, but are only mildy significant.

LDA reaches an earlier minimum for perplexity and then it usually increases, though PMI does increase as well. Models like HDP-LDA and NP-LDA usually keep on improving in PMI as the number of topics increases and hold-out perplexity often waivers about, gradually increasing after a later minimum. For instance, for the small MLT data set they reach a minimum perplexity at about K=20. All the while, PMI keeps improving. For data sets like Reuters-21578 a much larger number of topics can be supported, for instance $K > 500$ easily. The eventual increase in perplexity for larger K seems counter-intuitive given the non-parametric slogan of "estimating the right dimension of the model from the data". However, remember, we have initialization artifacts to deal with. Initializing with substantially too many topics leads to fragmentation/duplication of the topics not subsequently handled by simple Gibbs sampling. To deal with this sort of affect, we need something like split-merge operators in the sampler [2].

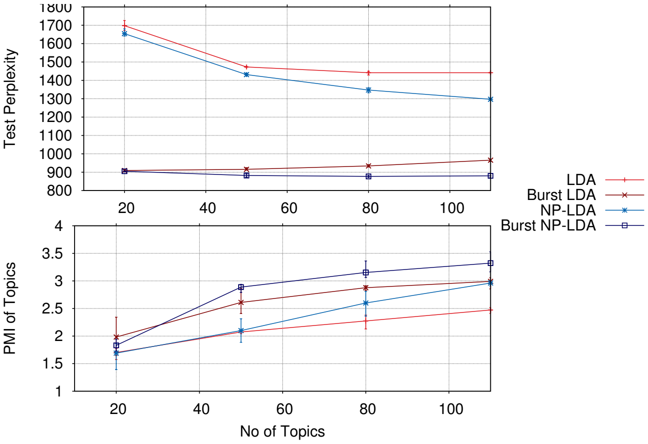


Figure 4: Perplexity and PMI on the RML data for LDA, Burst LDA and NP-LDA, Burst NP-LDA.

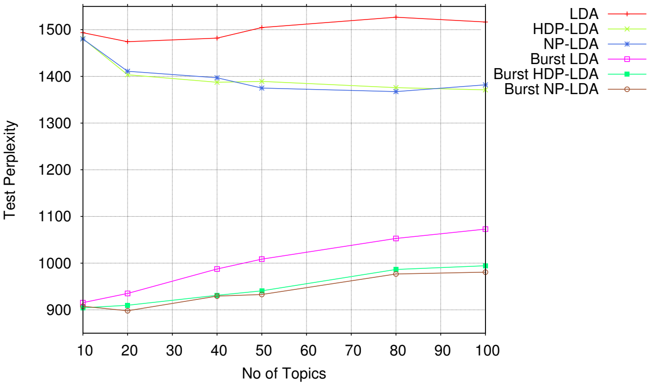


Figure 5: Perplexity as the number of topics (K) changes for different algorithms on the MLT data.

With the burstiness models, however, the change in performance is dramatic. Burstiness almost always improves PMI, sometimes substantially and the drop in perplexity is always dramatic. Burstiness makes perplexity peak much earlier w.r.t. the number of topics, but for the non-parametric models the subsequent rise in perplexity thereafter is mild. The non-parametric models cope better with the challenges of sampling behind the bursty front-end. However the best perplexity is reached for low number of topics on MLT where all the different models (LDA, HDP-LDA, NP-LDA) have similar perplexity. For the larger RML data set, LDA's perplexity again peaks earlier but NP-LDA's keeps improving for the number of topics considered.

5.3 Performance Comparisons

This section compares `hca` performance with previous algorithms.

5.3.1 *Comparison with* `onlinehdp` *and* `HDP`

In order to compare the different systems, `hca` versus `onlinehdp` and `HDP` we use the RML and TNG data. We used a fixed set of hyperparameters with no sampling so all discount parameters are set to 0 and the relevant concentration parameters set to 1 (b_α, b_θ) and a symmetric $\beta = (0.01)\vec{1}$ is used. For `onlinehdp` we did a large number of runs vary-

Table 4: Document completion perplexity and PMI for `hca` variants. Data is presented as "Perplexity/PMI". "HDP" is short for "HDP-LDA".

Data (K)	LDA	Burst LDA	HDP	Burst HDP	NP-LDA	Burst NP-LDA
MLT(10)	1493.62/2.33	915.46/2.47	1480.85/2.61	**904.29**/2.59	1480.20/2.38	907.74/**2.70**
MLT(50)	1504.63/2.94	1008.68/3.26	1389.29/3.70	940.69/3.63	1375/3.47	**932.88/3.93**
RML(50)	1472.87/2.07	915.65/2.61	1427.28/2.25	891.07/2.73	1431.29/2.10	**882.06/2.89**
RML(110)	1441.55/2.43	965.56/2.99	1308.83/3.05	889.42/3.31	1297.08/2.96	**880.22/3.32**
PN(160)	4232.08/3.69	2988.69/4.18	3801.42/4.50	2689.19/4.62	3785.05/4.39	**2657.78/4.70**
PN(240)	4306.63/4.07	3081.19/4.45	3726.05/4.75	**2720.98**/4.76	3676.35/4.72	2734.66/**4.78**

ing $\tau = 1, 4, 16, 64$. $\kappa = 0.5, 0.8$ and $K = 150, 300$ and $batchsize = 250, 1000$. Note $\tau = 64, \kappa = 0.8$ are recommend in [27]. Only the fastest and best converging result is given for `onlinehdp`. We did one run of both `hca` and `HDP` with these settings noting that the differences are way outside of the range of typical statistical variation between individual runs. Plots of the runs over time are given in Figures 6 and 7. The final PMI scores for the 3 algorithms are given in Table 5.

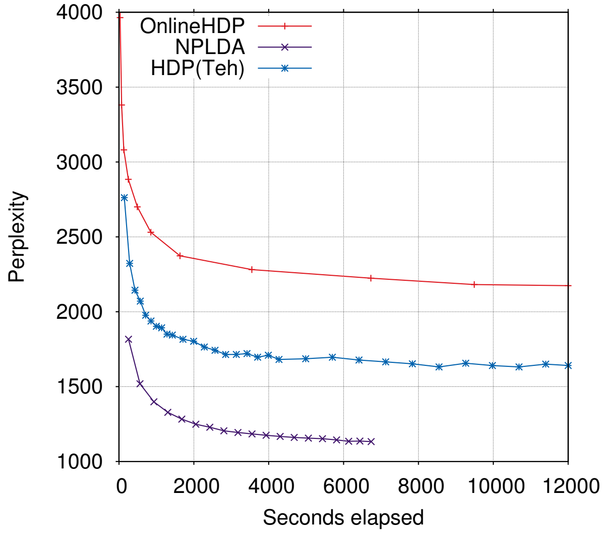


Figure 6: Comparative perplexity for one run on the RML data.

Table 5: PMI scores for the comparative runs.

	`onlinehdp`	`hca`	`HDP`
RML	2.607	3.47	4.452
TNG	4.042	4.017	4.887

Table 6: Effective Number of Topics for the comparative runs.

	`onlinehdp`	`hca`	`HDP`
RML	37.0	155	149
TNG	7.1	92.7	89.6

The improvement in perplexity of `hca` over `HDP` is not that surprising because comparative experiments on even simple models show the significant improvement of table indicator methods over CRP methods [6], and Sato et al. [20] also report substantial differences between different formulations for variational HDP-LDA. However, the poor performance of `onlinehdp` needs some explanation. On looking at the topics discovered by `onlinehdp`, we see there are many duplicates. Moreover, the topic proportions given by the $\vec{\alpha}$ vector show extreme bias towards earlier topics. It is known, for instance, that variational methods applied to the Dirichlet make the probability estimates more extreme. In this model one is working with a tree of Betas, so it seems the effect is confounded. A useful diagnostic here is the "Effective Number of Topics" which is given by exponentiating the entropy of the estimated $\vec{\alpha}$ vector, shown in Table 6. One can see `hca` and `HDP` are similar here but `onlinehdp` has a dramatically reduced number of topics. The non-duplicated topics in the `onlinehdp` result, however, look good in terms of comprehensability, so the online stochastic variational method is clearly a good way to get a smaller number of topics from a very large data set.

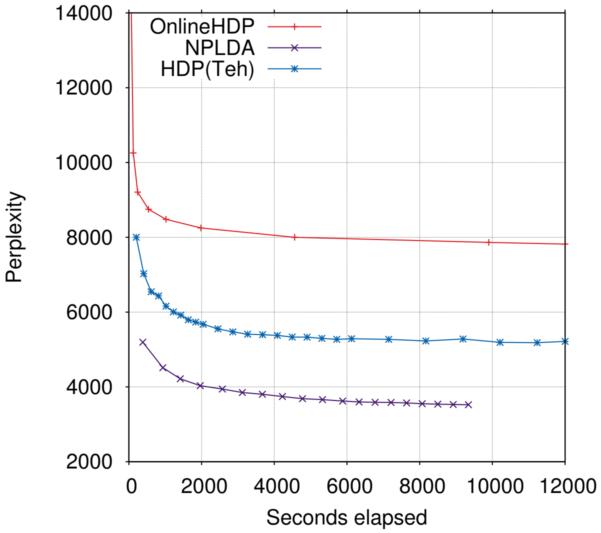


Figure 7: Comparative perplexity for one run on the TNG data.

5.3.2 *Comparison with* `Mallet`

`Mallet` suppports asymmetric-symmetric LDA, which is a form of truncated HDP-LDA using finite symmetric Dirichlet to truncate a GEM. We compare the implementation of HDP-LDA in `Mallet` and `hca`. Results are reported for

Table 7: Comparative Results for Mallet.

		hca	
Dataset(K)	Mallet	(HDP-LDA)	(NP-LDA)
RML(300)	1404 ± 8	1280 ± 2	1145 ± 2
TNG(300)	4081 ± 27	3999 ± 10	3586 ± 8
MLT(50)	1357 ± 14	1389 ± na	1375 ± na
PN(240)	3844 ± 24	3726 ± na	3676 ± na

Table 8: Comparative Results for PCVB0

		hca	
K	PCVB0	(HDP-LDA)	(NP-LDA)
200	1285 ± 10	1267 ± 5	1193 ± 5
300	1275 ± 10	1223 ± 5	1151 ± 5

"RML" and "TNG" datasets with 300 topics as per previous, and also some from Table 4. As suggested in [16] we run `Mallet` for 2000 iterations, and optimise the hyperparameters every 10 major Gibbs cycles after an initial burn-in period of 50 cycles, to get the best results. Table 7 presents the comparative results. We can see that `hca` generally produces better results. Note that results produced by the full asymmetric version NP-LDA are even better, an option not implemented in `Mallet`.

5.3.3 *Comparison with PCVB0*

We also sought to compare `hca` with the variants of `PCVB0` reported in [20]. These are a family of simplified variational algorithms, though the different variants seem to perform similarly. Without details of the document pre-processing, it was difficult to reproduce comparable datasets. Thus only results for their "KOS blog corpus," available preprocessed from the UCI Machine Learning Repository, where used in producing the comparisons presented in Table 8. We note the smaller difference here in perplexity is such that better hyper-parameter estimation with PCVB0 could well make the algorithms more equal. Interestly, Sato et al. report little difference between the symmetric or asymmetric priors on the Dirichlet on $\vec{\phi}_k$. In contrast, our corresponding asymmetric version NP-LDA shows significant improvements.

5.3.4 *Comparison on NIPS 1988-2000 Dataset*

A split-merge variant of HDP-LDA has been developed [2] that was compared with online and batch variational algorithms. For the NIPS data they have made runs with $K = 300$ and they estimate all hyperparameters. They use a 80-20% split for document completion and we replicated the experiment with the same dataset, parameter settings and sampling. The results are show in Figure 8 and should be compared with [2, Figure 2(b)]. Their results show plots for 40 hours whereas we ran for 4.5 hours, so our algorithm is approximately 4 times faster per document. Our Gibbs implementation of HDP-LDA substantially beats all other non-split-merge algorithms. Not surprisingly, the sophisticated split-merge sampler eventually reaches the performance of ours. Note the NP-LDA model is superior to HDP-LDA on this data, and the bursty versions are clearly superior to all others.

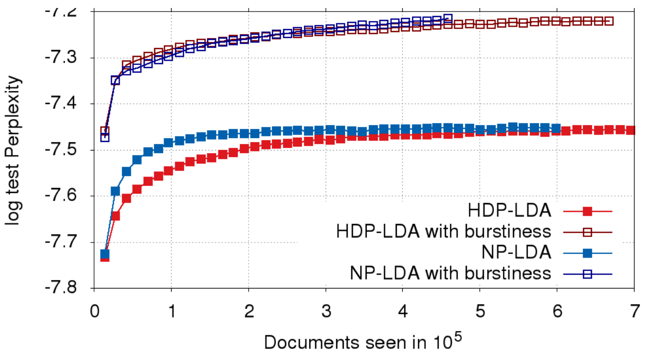


Figure 8: Convergence on Roweis' NIPS data for $K = 300$.

5.4 Effect of Hyperparameters on the Number of Topics

Standard reporting of experiments using HDP-LDA usually sets the β parameter which governs the symmetric prior for the $\vec{\phi}_k$. For instance, some authors [13] call this η and it is set to 0.01. Here we explore what happens when we vary this parameter for the RML data. Note we have done this experiment on most of the data sets and the results are comparable. We train HDP-LDA for 1000 Gibbs cycles and then record the evaluation measures. This takes 60 minutes on the desktop for each value of β. We also do a run where β is sampled. For each of the curves, the stopping point on the right gives the number of full topics used by the algorithms (ignoring trivially populated topics with 1-2 words). So the lowest perplexity is achieved by HDP-LDA with $\beta = 0.001$ where roughly $K = 2,400$ topics are used. Sampling β roughly tracks the lowest achieved for each number of topics.

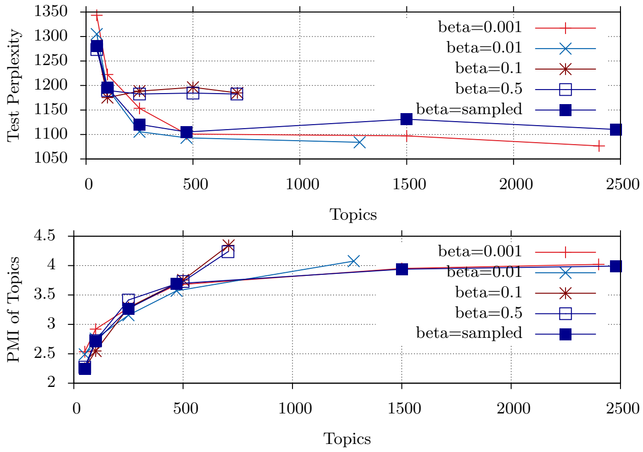


Figure 9: Perplexity and PMI for the RML data when varying β in the symmetric prior for HDP-LDA.

The PMI results also indicate that for larger β one obtains more comprehensible topics, though less of them. Thus there is a trade-off: if you want less but more comprehensible topics, for instance a coarser summary of the content, then make β larger. If you want a better fit to the data, or more finely grained topics, then estimate β properly.

Table 9: Low proportion topics (proportion below 0.001) with lower variance factor for LAT data when $K = 500$.

PMI	topic words
0.31	Zsa gabor capos slapping avert anhalt enright rolls-Royce cop-slapping Hensley judgeship Leona
2.32	herald tribune examiner dailies gannet batten numeric press-telegram petersburg sentinel
4.02	Baker PT evangelist bakers Tammy Faye swagged evangelists televangelists defrocked

Thus we can see that the number of topics found by HDP-LDA is significantly affected by the hyper parameter β, and thus it is probably inadvisable to fix it without careful experimentation, consideration or sampling. Moreover, the number of topics on RML, with roughly 20,000 documents is up to 2,000. Inspection shows a good number of these are comprehensible. With larger collections we claim it would be impractical to attempt to "estimate" the right number of topics. For larger collections, one could be estimating tens of thousands of topics. Is this large number of topics even useful?

5.5 Topic Specific Concentrations

For the topic burstiness model of Figure 3 we had topic specific concentrations to the PYP, $b_{\phi,k}$. Now the concentration and discount together control the variance. So for document i and topic k, the variance of a word probability $\psi_{i,k,w}$ from its mean $\phi_{k,w}$ will be $\left(\frac{1-a_{\phi}}{1+b_{\phi,k}}\right)\phi_{k,w}$ [3]. We call the ratio the variance factor. If it is close to one then the word proportions $\vec{\psi}_{i,k}$ for the topic have little relationship to their mean $\vec{\phi}_k$. If close to zero they are similar. Figure 10 considers 500 topics from a model built on the LAT data with $K = 500$ using PYP-LDA and topic burstiness. About 15% of the topics have low values for concentration that make the topics effectively random, and thus not properly used. Examples of topics with low proportions but variance factor below 0.4, so the topics are still use able, are given in Table 9. The first topic is actually about two issues: the first is the Zsa Zsa Gabor slapping incident, and the second is about Orange County Dist. Atty.s Avert and Enright.

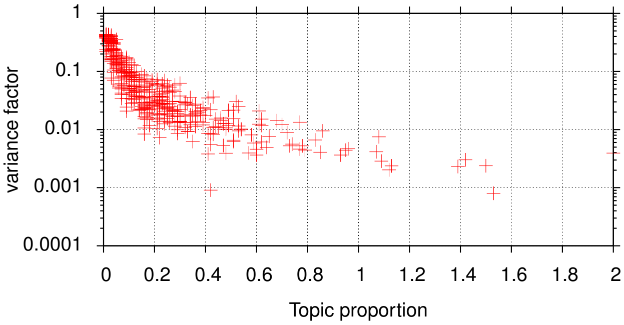


Figure 10: Topic proportions versus the variance factor for LAT data when $K = 500$.

6. CONCLUSION

We have shown that an implementation of the HDP-LDA non-parametric topic model and related non-parametric extension NP-LDA using block table indicator Gibbs sampling methods [4] are significantly superior to recent variational methods in terms of perplexity of results. The NP-LDA is also significantly superior in perplexity to the `Mallet` implementation of truncated HDP-LDA (masquerading as asymmetric symmetric LDA). Taking account of the different implementation languages, the newer Gibbs samplers and variational methods also have the same memory footprint. `Mallet` is substantially faster, however, and performs well for HDP-LDA.

We note that these have two goals, (A) better estimating prior topic or word proportions, and (B) estimating the "right" number of topics. The non-parametric methods seem superior at the first goal (A) over the parametric equivalents. Given that the estimated number of topics grows substantially with the collection sizes, it is not clear how important goal (B) can be. Arguably, goal (A) is the more important one.

Moreover, we have developed a Gibbs theory of burstiness that:

- Is implemented as a front-end so can in principle readily be applied to most variants of a topic model that use a Gibbs sampler.
- It is a factor of 1.5-2 slower per major Gibbs cycle.

This will allow the wide variety of topic-model variants to easily take advantage of the burstiness model.

Through the experiments, we have illustrated some characterizations of the models, for instance:

- Our asymmetric-asymmetric NP-LDA model is about 75% slower than HDP-LDA but generally performs better than HDP-LDA, a different result to published results [25, 20] due to the different algorithms.
- The topic comprehensibility (as measured using PMI) is substantially improved by the burstiness version, as reported in the original work [5].
- The topic concentration parameter in the burstiness model goes very low when the topic is insignificant. We can use this to estimate which topics have become inactive in the model.
- The concentration parameter for the topic-word vectors significantly affects results, so care should be taken in experiments using these models.

7. ACKNOWLEDGEMENTS

Both authors were funded partly by NICTA. NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program. Thanks to Changyou Chen and Kar Wai Lim for their feedback and Changyou for running the `HDP` experiments.

8. REFERENCES

[1] J. Boyd-Graber, D. Blei, and X. Zhu. A topic model for word sense disambiguation. In *EMNLP-CoNLL*, pages 1024–1033, 2007.

[2] M. Bryant and E. Sudderth. Truly nonparametric online variational inference for hierarchical Dirichlet processes. In P. Bartlett, F. Pereira, C. Burges,

L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2708–2716. 2012.

[3] W. Buntine and M. Hutter. A Bayesian view of the Poisson-Dirichlet process. Technical Report arXiv:1007.0296 [math.ST], arXiv, Feb. 2012.

[4] C. Chen, L. Du, and W. Buntine. Sampling table configurations for the hierarchical Poisson-Dirichlet process. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD*, pages 296–311. Springer, 2011.

[5] G. Doyle and C. Elkan. Accounting for burstiness in topic models. In *Proc. of the 26th Annual Int. Conf. on Machine Learning*, ICML '09, pages 281–288, 2009.

[6] L. Du. *Non-parametric Bayesian Methods for Structured Topic Models A Mixture Distribution Approach.* PhD thesis, School of Computer Science, the Australian National University, Canberra, Australia, 2011.

[7] L. Du, W. Buntine, and H. Jin. Modelling sequential text with an adaptive topic model. In *Proc. of the 2012 Joint Conf. on EMNLP and CoNLL*, pages 535–545. ACM, 2012.

[8] L. Du, W. Buntine, and M. Johnson. Topic segmentation with a structured topic model. In *HLT-NAACL*, pages 190–200. The Association for Computational Linguistics, 2013.

[9] L. Du, W. Buntine, and M. Johnson. Topic segmentation with a structured topic model. In *Proceedings of NAACL-HLT*, pages 190–200, 2013.

[10] W. R. Gilks and P. Wild. Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*, pages 337–348, 1992.

[11] T. Griffiths and M. Steyvers. Finding scientific topics. *PNAS Colloquium*, 2004.

[12] S. Harter. A probabilistic approach to automatic keyword indexing. Part II. An algorithm for probabilistic indexing. *Jnl. of the American Society for Information Science*, 26(5):280–289, 1975.

[13] M. Hoffman, D. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

[14] H. Ishawaran and L. James. Generalized weighted Chinese restaurant processes for species sampling mixture models. *Statistica Sinica*, 13:1211–1235, 2003.

[15] H. Ishwaran and L. James. Gibbs sampling methods for stick-breaking priors. *Journal of ASA*, 96(453):161–173, 2001.

[16] A. K. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[17] D. Newman, J. Lau, K. Grieser, and T. Baldwin. Automatic evaluation of topic coherence. In *Proc. of the 2010 Annual Conf. of the NAACL*, pages 100âĂŞ108, 2010.

[18] S. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, Apr. 2009.

[19] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proc. of the 20th Annual Conf. on Uncertainty in Artificial Intelligence (UAI-04)*, pages 487–49, 2004.

[20] I. Sato, K. Kurihara, and H. Nakagawa. Practical collapsed variational Bayes inference for hierarchical Dirichlet process. In *Proc. of the 18th ACM SIGKDD international conf. on Knowledge discovery and data mining*, pages 105–113. ACM, 2012.

[21] I. Sato and H. Nakagawa. Topic models with power-law using Pitman-Yor process. KDD '10, pages 673–682. ACM, 2010.

[22] Y. Teh, K. Kurihara, and M. Welling. Collapsed variational inference for HDP. In *NIPS '07*. 2007.

[23] Y. W. Teh. A Bayesian interpretation of interpolated Kneser-Ney. Technical Report TRA2/06, School of Computing, National University of Singapore, 2006.

[24] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the ASA*, 101(476):1566–1581, 2006.

[25] H. Wallach, D. Mimno, and A. McCallum. Rethinking LDA: Why priors matter. In *Advances in Neural Information Processing Systems 19*, 2009.

[26] H. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *ICML '09*, pages 672–679. 2009.

[27] C. Wang, J. Paisley, and D. Blei. Online variational inference for the hierarchical Dirichlet process. In *AISTATS '11*. 2011.