Sequential Latent Dirichlet Allocation: Discover Underlying Topic Structures within a Document

Lan Du*[†], Wray Buntine[†]* and Huidong Jin[‡]*
* *CECS,The Australian National University, Canberra, Australia*
[†] *National ICT Australia, Canberra, Australia*
[‡]*CSIRO Mathematics, Informatics and Statistics, Canberra, Australia*
Email: {Lan.Du, Wray.Buntine}@nicta.com.au; Warren.Jin@csiro.au

***Abstract*—Understanding how topics within a document evolve over its structure is an interesting and important problem. In this paper, we address this problem by presenting a novel variant of Latent Dirichlet Allocation (LDA): Sequential LDA (SeqLDA). This variant directly considers the underlying sequential structure, *i.e.*, a document consists of multiple segments (*e.g.*, chapters, paragraphs), each of which is correlated to its previous and subsequent segments. In our model, a document and its segments are modelled as random mixtures of the same set of latent topics, each of which is a distribution over words; and the topic distribution of each segment depends on that of its previous segment, the one for first segment will depend on the document topic distribution. The progressive dependency is captured by using the nested two-parameter Poisson Dirichlet process (PDP). We develop an efficient collapsed Gibbs sampling algorithm to sample from the posterior of the PDP. Our experimental results on patent documents show that by taking into account the sequential structure within a document, our SeqLDA model has a higher fidelity over LDA in terms of perplexity (a standard measure of dictionary-based compressibility). The SeqLDA model also yields a nicer sequential topic structure than LDA, as we show in experiments on books such as Melville's "The Whale".**

***Keywords*-Latent Dirichlet Allocation, Poisson-Dirichlet process, collapsed Gibbs sampler, document structure**

I. Introduction

Probabilistic topic modelling, a methodology for reducing high dimensional data vectors to low dimensional representations, has a successful history in exploring and predicting the underlying semantic structure of documents, based on a hierarchical Bayesian analysis of the original texts [1], [2]. Its fundamental idea is that documents can be taken as mixtures of topics, each of which is a probability distribution over words, under the "*bag-of-words*" assumption.

Nowadays, topic modelling has been receiving increasing attention in both data mining and machine learning communities. A variety of topic models have been developed to analyze the content of documents and the meaning of words. These include models of words only [2], of topic trends over time [3]–[6], of word-order with Markov dependencies [7], of words and supervised information, *e.g.*, authors [8], class labels [9], of the intra-topic correlation (*i.e.*, the hierarchical structure of topics) [10], [11], of segments in documents [12], and so on. Although assumptions made by these models are slightly different, they share the same general format: mixtures of topics, probability distributions over words and hierarchical graphical model structures.

Many documents in corpora come naturally with structure. They consists of meaningful segments (*e.g.*, chapters, sections, or paragraphs), each containing a group of words, *i.e.*, document-segment-word structure. For instance, an article has sections; a novel has chapters; and these themselves contain paragraphs, each of which is also composed of sentences. Thus, a different challenge in text mining is the problem of understanding the document structure.

With reference to the way in which people normally compose documents, each document will have a main idea, and its segments should be associated with some ideas that we call sub-ideas. These kinds of ideas expressed in the document do not occur in isolation. They should be well structured, accessible and understandable to the reader. As we read and interpret documents, we should bear in mind correlations between main idea and sub-ideas, and correlations between sub-ideas of adjacent segments. Therefore, we believe segments not only have meaningful content but also provide contextual information for subsequent segments.

Can we statistically analyze documents by explicitly modelling the document structure in a sequential manner? We adopt probabilistic generative models called topic models to test this hypothesis. Thus, the main idea of a document and sub-ideas of its segments can be modelled here by the distributions over latent topics. However most of the existing topic models are not aware of the underlying document structure. They only consider one level, *i.e.*, document-words. Although the Latent Dirichlet Co-Clustering (LDCC) Model [12], as shown in Figure 1(b), has taken into consideration the segmented structure of a document, the authors ignore the topical dependencies between segments, and those between segments and the whole document. Griffiths et al. [7] proposed a model that captures both syntactic (*i.e.*, word-order) and semantic (*i.e.*, topic) dependencies, by using Markov dependencies and topic model, respectively. But, the topical dependency buried in higher level of document structure is also true and exists.

1550-4786/10 $26.00 © 2010 IEEE
DOI 10.1109/ICDM.2010.51

Different from previous topic models, this paper presents a novel variant of the Latent Dirichlet Allocation (LDA) model [2], a topic model, called Sequential Latent Dirichlet Allocation (SeqLDA), that explicitly models the underlying document structure. Although in this work, we have restricted ourselves to the study of the sequential topic structure of a document, that is how a sub-idea is closely related to the preceding and subsequent segments. The progressive topical dependency is captured using a nested extension of the two-parameter Poisson-Dirichlet process (PDP) [13], [14], based on recent theoretical results in finite discrete spaces [15]. The nested PDP is defined as $\boldsymbol{u}_i \sim PDP(a_i, b_i, \boldsymbol{u}_{i-1})$, where a_i is a discount parameter, b_i is a strength parameter, and $\boldsymbol{u}_{i-1}$ is base distribution for $\boldsymbol{u}_i$ in a recursive fashion, as those in [16]. The advantage of using the PDP in a nested way is that it allows us to integrate out the real valued parameters, *i.e.*, the PDP is self-conjugate. We also develop here a collapsed Gibbs sampler for this nested case that provides an efficient and space compact sampler for the PDP.

Considering this sequential structure, we can explore how topics are evolving among, for example, paragraphs in an essay, or chapters in a novel; and detect the rising and falling of a topic in prominence. The evolvement can be estimated by exploring how the topic proportion changes in segments. Obviously, tackling topic modelling together with the topical structures buried in documents provides a solution for going beyond the "*bag-of-words*" assumption.

Existing studies about topic evolvements or trends focus mainly on time series, *i.e.*, topics over time, rather than those hiding inside each document. They explore how topics change, rise and fall, by taking into account timestamps and other information (*e.g.*, citations) associated with documents. Blei and Lafferty [3] propose a dynamic topic model (DTM) to capture the topic evolution in document collections that are organized sequentially into several discrete time periods, and then within each period an LDA model is trained on its documents. Wang and McCallum [4] present another topic over time (ToT) model, a non-Markov continuous time topic model. Instead of training an LDA model for documents in each time stamp, ToT assumes that words and timestamps can be jointly generated by latent topics. Indeed, timestamps in ToT are treated as supervised information. Leveraging the citation information, He et al. [17] develop inheritance topic model to understand topic evolution. Significantly, the difference between these models and our SeqLDA model is that, instead of modelling topic trends in document collections based on documents' timestamps, we model topic progress within each individual document by capitalizing on the correlations among its segments, *i.e.*, the underlying sequential topic structure, according to the original document layout. Moreover, compared to [3], the Markov dependencies in our model are put on distributions over topics, rather than distributions over words. In such a way, we can directly model the topical dependency between a segment and its successor.

The rest of the paper is organized as follows. In Section II we describe the SeqLDA model in detail, and compare it with the LDA Model. Section III elaborates an efficient collapsed Gibbs sampling algorithm for SeqLDA. Experiment results are reported in Section IV. Section V gives a brief discussion and concluding comments.

II. Sequential Latent Dirichlet Allocation

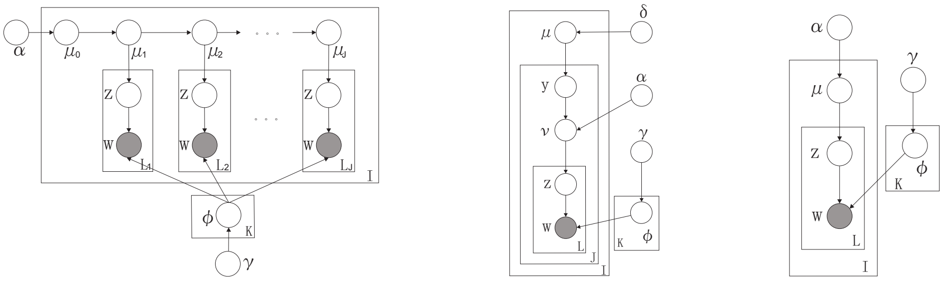
In this section, we present the novel Sequential Latent Dirichlet Allocation model (SeqLDA) which models how topics evolve among segments in a document. We assume that there could be some latent sequential topic structures within each document, *i.e.*, the ideas within a document evolve smoothly from one segment to another, especially in novels and many books. This assumption intuitively originates from the way in which people normally structure ideas in their writing. Before specifying the SeqLDA model, we list notations and terminologies used in this paper. Notations are depicted in Table I. We define the following terms:

- A *word* is the basic unit of our data, selected from a vocabulary indexed by $\{1, \cdots, W\}$.
- A *segment* is a group of L words. It can be a chapter, section, or paragraph.
- A *document* is a sequence of J segments.
- A *corpus* is a collection of I documents.

The basic idea of our model is to assume that each document i is a certain mixture of latent topics, denoted by the distribution $\boldsymbol{\mu}_{i,0}$, and is composed of a sequence of meaningful segments; each of these segments also has a mixture over the same set of latent topics as those for the document, and these are indicated by distribution $\boldsymbol{\mu}_{i,j}$ for segment j. Note that the index of a segment complies with its position in the original document layout, which means the first segment is indexed by $j = 1$, the second segment is indexed by $j = 2$, and so on. Both the main idea of a document and the sub-ideas of its segments are modelled here by these distributions over topics.

Table I
List of notations

Notation.	Description.
K	number of topics
I	number of documents
J_i	number of segments in document i
$L_{i,j}$	number of words in document i, segment j
W	number of words in dictionary
$\boldsymbol{\alpha}$	K-dimensional vector for the Dirichlet prior for document topic distributions
$\boldsymbol{\mu}_{i,0}$	document topic distribution for document i
$\boldsymbol{\mu}_{i,j}$	segment topic distribution for segment j in document i
$\boldsymbol{\Phi}$	word probability vectors as a $K \times W$ matrix
$\boldsymbol{\phi}_k$	word probability vector for topic k, entries in Φ
$\boldsymbol{\gamma}$	W-dimensional vector for the Dirichlet prior for each $\boldsymbol{\phi}_k$
$w_{i,j,l}$	word in document i, segment j, at position l
$z_{i,j,l}$	topic for word in document i, segment j, at position l



(a) SeqLDA

(b) LDCC

(c) LDA

Figure 1. Graphical model representations for the SeqLDA model, the LDCC model and the LDA model

The development of a sequential structural generative model according to the above idea is based on nested PDPs, and models how the sub-idea of a segment is correlated to its preceding and following segments. Specifically, the correlation is simulated by the progressive dependency among topic distributions. That is, the j^{th} segment topic distribution $\boldsymbol{\mu}_{i,j}$ is the base distribution of the PDP for drawing the $(j+1)^{th}$ segment topic distribution $\boldsymbol{\mu}_{i,j+1}$; for the first segment, we draw its topic distribution $\boldsymbol{\mu}_{i,1}$ from the PDP with document topic distribution $\boldsymbol{\mu}_{i,0}$ as the base distribution. The strength parameter b and discount parameter a control the variation between the adjacent topic distributions. Figure 1(a) shows the graphical representation of the SeqLDA model. Shaded and unshaded nodes indicate observed and latent variables respectively. An arrow indicates a conditional dependency between variables, and plates indicate repeated sampling.

In terms of a generative process, the SeqLDA model can be also viewed as a probabilistic sampling procedure that describe how words in documents can be generated based on the latent topics. It can be depicted as follows: Step 1 samples the word distribution for topics, and Step 2 samples each document by breaking it up into segments:

1) For each topic k in $\{1, \ldots, \mathrm{K}\}$
   a) Draw $\boldsymbol{\phi}_k \sim Dirichlet_W(\boldsymbol{\gamma})$
2) For each document i
   a) Draw $\boldsymbol{\mu}_{i,0} \sim Dirichlet_K(\boldsymbol{\alpha})$
   b) For each segment $j \in \{1, \ldots, J_i\}$
      i) Draw $\boldsymbol{\mu}_{i,j} \sim PDP(a, b, \boldsymbol{\mu}_{i,j-1})$
      ii) For each word $w_{i,j,l}$, where $l \in \{1, \ldots, L_{i,j}\}$
         A) draw $z_{i,j,l} \sim multinomial_K(\boldsymbol{\mu}_{i,j})$
         B) draw $w_{i,j,l} \sim multinomial_W(\boldsymbol{\phi}_{z_{i,j,l}})$

We have assumed the number of topics (*i.e.*, the dimensionality of the Dirichlet distribution) is known and fixed, and the word probabilities are parameterized by a $K \times W$ matrix $\boldsymbol{\Phi} = (\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_K)$, and will be estimated though the learning process. $\boldsymbol{\mu}_{i,0}$ is sampled from the Dirichlet distribution with prior $\boldsymbol{\alpha}$, and others are sampled from the PDP. Both the Dirichlet distribution and the PDP are conjugate priors for the multinomial distribution, and the PDP is also self-conjugate in a sense. Choosing these conjugate priors makes the statistical inference easier, as discussed in the next section. The joint distribution of all observed and latent variables can be constructed directly from Figure 1(a) using the distributions given in the above generative process, as below:

$$\begin{aligned} & p(\boldsymbol{\mu}_{i,0}, \boldsymbol{\mu}_{i,1:J}, \boldsymbol{z}, \boldsymbol{w} | \boldsymbol{\alpha}, \boldsymbol{\Phi}, a, b) \\ = \; & p(\boldsymbol{\mu}_{i,0}|\boldsymbol{\alpha}) \prod_{j=1}^{J_i} p(\boldsymbol{\mu}_{i,j}|a, b, \boldsymbol{\mu}_{i,j-1}) \\ & \prod_{l=1}^{L_j} p(z_{i,j,l}|\boldsymbol{\mu}_{i,j}) p(w_{i,j,l}|\boldsymbol{\phi}_{z_{i,j,l}}) \end{aligned}$$

where $p(\boldsymbol{\mu}_{i,j}|a, b, \boldsymbol{\mu}_{i,j-1})$ is given by $PDP(a, b, \boldsymbol{\mu}_{i,j-1})$.

From the notion of the proposed model, we can find the obvious distinction between the SeqLDA model and the LDA model (shown in Figure 1(c)): the SeqLDA model takes into account the sequential structure information of each document, *i.e.* the position of each segment that the LDA model ignores. Our SeqLDA model aims to capitalize on the information conveyed in the document layout, to explore how topics evolve within a document, and further to assist in understanding the original text. Although the LDA model can also be applied to segments directly, the progressive topical dependency between two adjacent segments would be lost by treating segments independently. Similar to the LDA model, the LDCC model [12], as shown in Figure 1(b), still has an implicit assumption that segments within a document are exchangeable, which is not always appropriate.

Therefore, if documents indeed have some latent sequential structures, considering this dependency means a higher fidelity of SeqLDA over LDA and LDCC. However, if the correlation among sub-ideas of segments is not obvious, taking the topic distribution of the j^{th} segment as the base distribution of the $(j + 1)^{th}$ segment may mis-interpret the document topic structure. In this sense, the SeqLDA model may be a deficient generative model, but it is still a prominent model and remains powerful if the progressive dependency is dynamically changed by optimizing the strength and discount parameters (a and b) for each segment within each document. Though for simplicity, we fix a and b for each document collection in all our experiments.

III. Inference Algorithm

In order to use the SeqLDA model, we need to solve the key inference problem which is to compute the posterior distribution of latent variables (*i.e.* topic distributions $\boldsymbol{\mu}_{i,0:J}$ and topic assignment $\boldsymbol{z}$) given the inputs (*i.e.* $\boldsymbol{\alpha}$, $\boldsymbol{\Phi}$, a and b) and observations $\boldsymbol{w}$. Unfortunately, this posterior distribution cannot be computed directly because of the intractable computation of marginal probabilities. As a consequence, we must appeal to approximate inference techniques, where some of the parameters (*i.e.* $\boldsymbol{\mu}_{i,0:J}$ and $\boldsymbol{\Phi}$ in our case) can be marginalized out, rather than explicitly estimated. In topic modeling literature, two standard approximation methods have often been used: variational inference [2] and Gibbs sampling [18]. Here, we pursue an alternative approximating strategy using the latter by taking advantage of the collapsed Gibbs sampler for the PDP [15].

Gibbs sampling is a special form of Markov chain Monte Carlo (MCMC) simulation which should proceed until the Markov chain has "converged" to its stationary state. Although, in practice, we run it for a fixed number of iterations. Collapsed Gibbs sampling capitalizes on the conjugacy of priors to compute the conditional posteriors. Thus, it always yields relatively simple algorithms for approximate inference in high-dimensional probability distributions by the stationary behavior of a Markov chain. Note that we use conjugate priors in our model, *i.e.* Dirichlet prior $\boldsymbol{\alpha}$ on $\boldsymbol{\mu}_{i,0}$ and $\boldsymbol{\gamma}$ on $\boldsymbol{\Phi}$, PDP prior on $\boldsymbol{\mu}_{i,j}$ (PDP is self-conjugate); thus $\boldsymbol{\mu}_{i,0:J}$ and $\boldsymbol{\Phi}$ can be integrated out.

In this section, we derive the collapsed Gibbs sampling algorithm for doing inference in the proposed model. Table II lists all the statistics required in our algorithm. Our PDP sampling is a collapsed version of what is known as the nested Chinese restaurant process (CRP) used as a component of different topic models [11]. The basic theory of the CRP and our collapsed version of it is summarized in Appendix A. The CRP model goes as follows: a Chinese restaurant has an infinite number of tables, each of which has infinite seating capacity. Each table serves a dish $k = 1, ..., K$, so multiple tables can serve the same dish. In modelling, we only consider tables which have at least one customer, called active tables. We have one Chinese restaurant for each segment in a document that models the topic proportions for the segment, and each restaurant serves up to K topics as dishes. The statistic $t_{i,j,k}$, called "table count", is introduced for the PDP in the CRP configuration [15], [16] and represents the number of active tables in the restaurant for segment i, j that are serving dish k. The table counts are treated as constrained latent variables that make it possible to design a collapsed Gibbs sampler. However, constraints hold on table counts: the total number of customers sitting at the $t_{i,j,k}$ tables serving dish k must be greater than or equal to the number of tables $t_{i,j,k}$.

Table II
List of statistics

Statistic.	Description.
$M_{i,k,w}$	topic by word total sum in document i, the number of words with dictionary index w and topic k.
$M_{k,w}$	$M_{i,k,w}$ totalled over documents i, *i.e.* $\sum_i M_{i,k,w}$
$\boldsymbol{M}_k$	vector of W values $M_{k,w}$
$n_{i,j,k}$	topic total in document i and segment j for topic k.
$N_{i,j}$	topic total sum in document i and segment j, *i.e.* $\sum_k n_{i,j,k}$.
$t_{i,j,k}$	table count in the CRP for document i and segment j, for topic k. This is the number of tables active for the k-th value.
$T_{i,j}$	total table count in the CRP for document i and segment j, *i.e.* $\sum_k t_{i,j,k}$.
$\boldsymbol{t}_{i,j}$	table count vector, *i.e.*, $(t_{i,j,1}, ..., t_{i,j,K})$ for segment j.
$u_{i,k}$	the smallest segment index j' in i, where $t_{i,j',k} = 0$.

A. The Model Likelihoods

To derive a collapsed Gibbs sampler for the above model, we need to compute the marginal distribution over the observation $\boldsymbol{w}$, the corresponding topic assignment $\boldsymbol{z}$, and the newly introduced latent variable, table counts $\boldsymbol{t}$. We do not need to include, *i.e.*, can integrate out, the parameter sets $\boldsymbol{\mu}_{i,0:J}$ and $\boldsymbol{\Phi}$, since they can be interpreted as statistics of the associations among $\boldsymbol{w}$, $\boldsymbol{z}$ and $\boldsymbol{t}$. Hence, we first recursively apply the collapsed Gibbs sampling function for the PDP, *i.e.* Eq. (6) in Appendix A, to integrating out the segment topic distributions $\boldsymbol{\mu}_{i,1:J}$; and then integrate out the document topic distributions $\boldsymbol{\mu}_{i,0}$ and the topic-word matrix $\boldsymbol{\Phi}$, as is usually done for collapsed Gibbs sampling in topic models. Finally, the joint conditional distribution of $\boldsymbol{z}_{1:I}$, $\boldsymbol{w}_{1:I}$, $\boldsymbol{t}_{1:I,1:J_i}$ can be computed as

$$
\begin{aligned}
& p(\boldsymbol{z}_{1:I}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I} \mid \boldsymbol{\Phi}, a, b) \qquad (1) \\
&= \prod_i \frac{\text{Beta}_K\left(\boldsymbol{\alpha} + \boldsymbol{t}_{i,1}\right)}{\text{Beta}_K\left(\boldsymbol{\alpha}\right)} \prod_{i,j} \frac{(b|a)_{T_{i,j}}}{(b)_{N_{i,j}+T_{i,j+1}}} \\
&\quad \prod_{i,j,k} S^{n_{i,j,k}+t_{i,j+1,k}}_{t_{i,j,k},a} \prod_k \frac{\text{Beta}_W\left(\boldsymbol{\gamma} + \boldsymbol{M}_k\right)}{\text{Beta}_W\left(\boldsymbol{\gamma}\right)}
\end{aligned}
$$

where $t_{i,j,k} \leq n_{i,j,k} + t_{i,j+1,k}$ and $t_{i,j,k} = 0$ *iff* $n_{i,j,k} + t_{i,j+1,k} = 0$; $\text{Beta}_K(\boldsymbol{\alpha})$ is K dimensional beta function that normalizes the Dirichlet; $(x)_N$ is given by $(x|1)_N$, and $(x|y)_N$ denotes the Pochhammer symbol with increment y, it is defined as

$$
\begin{aligned}
(x|y)_N &= x(x+y)...(x+(N-1)y) \\
&= \begin{cases} x^N & \text{if } y = 0 \\ y^N \times \frac{\Gamma(x/y+N)}{\Gamma(x/y)} & \text{if } y > 0 \ , \end{cases}
\end{aligned}
$$

where $\Gamma(\cdot)$ denotes the standard gamma function; $S^N_{M,a}$ is a generalized Stirling number given by the linear recursion [15], [16]: $S^{N+1}_{M,a} = S^N_{M-1,a} + (N - Ma)S^N_{M,a}$ for $M \leq N$. It is 0 otherwise and $S^N_{0,a} = \delta_{N,0}$. These numbers rapidly become very large so computation needs to be done in log space using a logarithmic addition.

B. The Collapsed Gibbs sampler

In each cycle of the Gibbs sampling algorithm, a subset of variables are sampled from their conditional distributions with the values of all the other variables given. In our case, the distributions that we want to sample from is the posterior distribution of topics ($\boldsymbol{z}$), and table counts ($\boldsymbol{t}$), given a collection of documents. Since the full joint posterior distribution is intractable and difficult to sample from, in each cycle of Gibbs sampling we will sample respectively from two conditional distributions: 1) the conditional distribution of topic assignment ($z_{i,j,l}$) of a single word ($w_{i,j,l}$) given the topics assignments for all the other words and all the table counts; 2) the conditional distribution of table count ($t_{i,j,k}$) of the current topic given all the other table counts and all the topic assignments.

In our model, documents are indexed by i, segments of each document are indexed by j according to their original layout, and words are indexed by l. Thus, with documents indexed by above method, we can readily yield a Gibbs sampling algorithm for the SeqLDA model as: for each word, the algorithm computes the probability of assigning the current word to topics from the first conditional distribution, while topic assignments of all the other words and table counts are fixed. Then the current word would be assigned to a sampled topic, and this assignment will be stored for being used when the Gibbs sampling cycles through other words. While scanning through the list of words, we should also keep track of the table counts for each segment. For each new topic that the current word is assigned to, the Gibbs sampling algorithm estimates the probabilities of changing the corresponding table count to different values by fixing all the topic assignments and all the other table counts. These probabilities are computed from the second conditional distribution. Then, a new value will be sampled and assigned to the current table count. Note that the values for the table count should be subject to some constraints that we will discuss in detail when we derive the two conditional distributions below.

Consequently, the aforementioned two conditional distributions we need to compute are, respectively,

1) $p\left(z_{i,j,l} = k \,|\, \boldsymbol{z}_{1:I} - \{z_{i,j,l}\}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i}, \boldsymbol{\alpha}, a, b\right)$
2) $p\left(t_{i,j,k} \,|\, \boldsymbol{z}_{1:I}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i} - \{t_{i,j,k}\}, \boldsymbol{\alpha}, a, b\right)$

where $z_{i,j,l} = k$ indicates the assignment of the l^{th} word in the j^{th} segment of document i to topic k, $\boldsymbol{z}_{1:I} - \{z_{i,j,l}\}$ presents all the topic assignments not including the l^{th} word, and $\boldsymbol{t}_{1:I,1:J_i} - \{t_{i,j,k}\}$ denotes all the table counts except for the current table count $t_{i,j,k}$. Following the CRP formulation, customers are words, dishes are topics and restaurants are segments in our case. All restaurants share a finite number of dishes, *i.e.*, K dishes. From Equation (1) and also seen from Equation (6) in the appendix, tables of $(j+1)^{th}$ restaurant are customers of j^{th} restaurant in nested CRPs. These counts have to comply with the following constraints: 1) $t_{i,j,k} = 0$ *iff* $n_{i,j,k} + t_{i,j+1,k} = 0$; 2) $t_{i,j,k} > 0$ *if* $n_{i,j,k} > 0$ or $t_{i,j+1,k} > 0$; 3) $n_{i,j,k} + t_{i,j+1,k} \geq t_{i,j,k} \geq 0$. For instance, the third constraint says that the total number of active tables serving dish k must be less than or equal to the total number of customers eating dish k. That is because each active table at least has one customer.

Input: a, b, $\boldsymbol{\alpha}$, $\boldsymbol{\gamma}$, K, $Corpus$, $MaxIteration$
Output: topic assignments for all words and all table counts

```
Topic assignment initialization: randomly initialize the topic assignment for all words.
Table count initialization: randomly initialize all $t_{i,j,k}$ s.t. $0 \leq t_{i,j,k} \leq n_{i,j,k} + t_{i,j+1,k}$
Compute all statistics listed in Table II
for $iter \leftarrow 1$ to $MaxIteration$ do
  foreach document $i$ do
    foreach segment $j$ in $i$, according to the original layout do
      foreach word $w_{i,j,l}$ in $j$ do
        Exclude $w_{i,j,l}$, and update the statistics with current topic $k' = z_{i,j,k}$ removed
        Look for the smallest $1 \leq j' \leq j$, s.t. $t_{i,j',k'} = 0$, and assign it to $u_{i,k'}$
        Sample new topic $k$ for $w_{i,j,l}$ using Eq. (2), Eq. (3) or Eq. (4) depending on the value of $u$
        Update the statistics with the new topic, and also update the value of $u_{i,k}$ if needed
        Remove the current table count $t'_{i,j,k}$ from the statistics
        Sample new table count $t_{i,j,k}$ for $k$ using Eq. (5)
        Update the statistics with the new table count
      end for
    end for
  end for
end for
```

Figure 2. Collapsed Gibbs sampling algorithm for the SeqLDA model

Then, considering the procedure of sampling a new topic for a word $w_{i,j,l}$, we need to remove the current topic (referred to as old topic) from the statistics. Assume the value of old topic $z_{i,j,l}$ is k, the number of words assigned to k in the j^{th} segment of document i, $n_{i,j,k}$, should decrease by one; then recursively check the table count $t_{i,j',k}$ for $1 \leq j' \leq j$ according to the constraints, and remove one if needed to satisfy the constraints, this check will proceed till somewhere the constraints hold; and finally assign the smallest j' to $u_{i,k}$ where the first constraint holds. Similarly, the same process should be done when assigning the current word to a new topic. We can prove, by recursion, that no $t_{i,j,k}$ goes from zero to non-zero or *vice versa* unless an $n_{i,j,k}$ does, so one only needs to consider the case where $n_{i,j,k} + t_{i,j+1,k} > 0$. Moreover, the zero $t_{i,j,k}$ forms a complete suffix of the list of segments, so $t_{i,j,k} = 0$ if and only if $u_{i,k} \leq j \leq J_i$ for some $u_{i,k}$.

Now, beginning with the conditional distribution, Eq. (1), using the chain rule, and taking into account all cases, we obtain the final full conditional distribution $p(z_{i,j,l} = k \,|\, \boldsymbol{z}_{1:I} - \{z_{i,j,l}\}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i}, \boldsymbol{\alpha}, a, b)$ with three different cases according to the value of $u_{i,k}$ as follows: when

$u_{i,k} = 1$, we have

$$\begin{aligned} & p(z_{i,j,l} = k \mid \boldsymbol{z}_{1:I} - \{z_{i,j,l}\}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i}, \boldsymbol{\alpha}, a, b) \\ = & \frac{\left(\alpha_k + t'_{i,1,k}\right)\left(b + aT'_{i,1}\right)}{\sum_k \alpha_k + \sum_k t'_{i,1,k}} \\ & \prod_{j'=2}^{j} \left(\frac{b + aT'_{i,j'}}{b + N_{i,j'-1} + T'_{i,j'}}\right) \frac{\gamma_{w_{i,j,l}} + M'_{k,w_{i,j,l}}}{\sum_w(\gamma_w + M'_{k,w})} \end{aligned} \tag{2}$$

When $1 < u_{i,k} \leq j$, the conditional probability is

$$\begin{aligned} & p(z_{i,j,l} = k \mid \boldsymbol{z}_{1:I} - \{z_{i,j,l}\}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i}, \boldsymbol{\alpha}, a, b) \\ = & \prod_{j'=u_{i,k}}^{j} \left(\frac{b + aT'_{i,j'}}{b + N_{i,j'-1} + T'_{i,j'}}\right) \\ & \frac{S^{n_{i,u_{i,k}-1,k}+1}_{t_{i,u_{i,k}-1,k},a}}{S^{n_{i,u_{i,k}-1,k}}_{t_{i,u_{i,k}-1,k},a}} \frac{\gamma_{w_{i,j,l}} + M'_{k,w_{i,j,l}}}{\sum_w(\gamma_w + M'_{k,w})} \end{aligned} \tag{3}$$

When $j < u_{i,k}$, it is simplified to

$$\begin{aligned} & p(z_{i,j,l} = k \mid \boldsymbol{z}_{1:I} - \{z_{i,j,l}\}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i}, \boldsymbol{\alpha}, a, b) \\ = & \frac{S^{n'_{i,j,k}+1+t_{i,j+1,k}}_{t'_{i,j,k},a}}{S^{n'_{i,j,k}+t_{i,j+1,k}}_{t'_{i,j,k},a}} \frac{\gamma_{w_{i,j,l}} + M'_{k,w_{i,j,l}}}{\sum_w(\gamma_w + M'_{k,w})} \end{aligned} \tag{4}$$

where the dash indicates statistics after excluding the current topics assignment $z_{i,j,l}$.

After sampling the new topic for a word, we need to stochastically sample the table count for this new topic, say k. Although we have summed out the specific seating arrangements (*i.e.* different tables and specific table assignments) of the customers in the collapsed Gibbs sampler for the PDP, we still need to sample how many tables are serving dish k (*i.e.* topic k in our model), given the current number of customers (*i.e.* words) eating dish k. The value of $t_{i,j,k}$ should be in the following interval: $\left[\max\left(1, t_{i,j-1,k} - n_{i,j-1,k}\right), n_{i,j,k} + t_{i,j+1,k}\right]$. Thus, given the current state of topic assignment of each word, the conditional distribution for table count $t_{i,j,k}$ can be obtained by similar arguments, as below.

$$\begin{aligned} & p(t_{i,j,k} \mid \boldsymbol{z}_{1:I}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i} - \{t_{i,j,k}\}, \boldsymbol{\alpha}, a, b) \\ \propto & \left(\frac{\Gamma\left(\alpha_k + t_{i,1,k}\right)}{\Gamma\left(\sum_k \alpha_k + \sum_k t_{i,1,k}\right)}\right)^{\delta_{j,1}} \\ & \left(\frac{S^{n_{i,j-1,k}+t_{i,j,k}}_{t_{i,j-1,k},a}}{(b)_{N_{i,j-1}+T'_{i,j}}}\right)^{1-\delta_{j,1}} (b|a)_{T'_{i,j}} S^{n_{i,j,k}+t_{i,j+1,k}}_{t_{i,j,k},a} \end{aligned} \tag{5}$$

The collapsed Gibbs sampling algorithm for our proposed model is outlined in Figure 2. We start this algorithm by randomly assigning words to topics in $[1, \ldots, K]$, and if the total number of customer, $n_{i,j,k} + t_{i,j+1,k}$, is greater than zero, the table count $t_{i,j,k}$ is initialized to 1. Each Gibbs sampler then continues applying Eq. (2), Eq. (3) or Eq. (4) to every word in the document collection; and applying Eq. (5) to each table count. A number of initial samples, *i.e.* samples before burn-in period, have to be discarded. After that, the Gibbs samples should theoretically approximate our target distribution (*i.e.* the posterior distribution of topics ($\boldsymbol{z}$), and table counts ($\boldsymbol{t}$)). Finally, we pick a number of Gibbs samples at regularly spaced intervals. In this paper, we average these samples to obtain the final sample, as done in [8]. This collapsed Gibbs sampling algorithm is easy to implement and requires little memory

IV. Experiment Settings and Results

We implemented the LDA model, the LDCC model and the SeqLDA model in C, and ran them on a desktop with Intel(R) Core(TM) Quad CPU (2.4GHz), though our code is not multi-threaded. Our previous comprehensive experimental results [19] on several well-known corpora as well as several patent document sets show that, though LDCC often outperforms LDA working on the document level, it performs quite similarly to LDA working on the segment level, in terms of document modelling. On the other hand, LDCC is not designed to uncover sequential topic structure either. Thus, we compare our SeqLDA directly with LDA working on both the document and the segment levels to facilitate easy comparison.

In this section, we first discuss the perplexity comparison between SeqLDA and LDA on a patent dataset by adopting the held-out method [8]. Then, we present topic evolvement analysis on two books, available at http://www.gutenberg.org. The former will show that our SeqLDA model is significantly better than LDA with respect to document modelling accuracy as measured by perplexity; and the latter will demonstrate the superiority of SeqLDA in topic evolvement analysis.

A. Data Sets

The patent dataset (*i.e.*, Pat-1000) has 1000 patents that are randomly selected from 8000 U.S. patents[1]. They are granted between Jan. and Apr. 2009 under the class "computing; calculating; counting". All patents are split into paragraphs according to the original layout in order to preserve the document structure. We remove all stop-words, extremely common words (*i.e.*, top 50), and less common words (*i.e.*, words appear in less than 5 documents). No stemming has been done. We here treat paragraphs as segments in the SeqLDA model. The two books we choose for topic evolvement analysis are "The Prince" by Niccolò Machiavelli and "The Whale" by Herman Melville, also known as "Moby Dick". They are split into chapters which are treated as segments, and only stop-words are removed. Table III shows the statistics of these datasets.

[1] All patents are from Cambia, http://www.cambia.org/daisy/cambia/home.html

Table III
DATASET STATISTICS

	The Prince	The Whale	Pat-1000	
			Training	Testing
No. documents	1	1	800	200
No. segments	26	135	49,200	11,360
No. words	10,705	88,802	2,048,600	464,460
Vocabulary	3,315	16,223	10,385	

Table IV
P-VALUES FOR PAIRED T-TEST FOR RESULTS IN FIGURE 3

	Pat-1000		
	SeqLDA	SeqLDA_D	SeqLDA_P
LDA_D	7.5e-4	3.3e-4	3.2e-5
LDA_P	3.0e-3	1.9e-2	3.6e-3

B. Document modelling

We first follow the standard way in document modelling to evaluate the per-word predicative perplexity of the SeqLDA model and the LDA model. The perplexity of a collection of documents is formally defined as: $exp\left\{-\frac{\sum_{i=1}^{I}\ln p(\boldsymbol{w}_i)}{\sum_{i=1}^{I}N_i}\right\}$, where $\boldsymbol{w}_i$ indicates all words and N_i indicates the total number of words in document i respectively. A lower perplexity over unseen documents means better generalization capability. In our experiments, it is computed based on the held-out method introduced in [8]. In order to calculate the likelihood of each unseen word in SeqLDA, we need to integrate out the sampled distributions (*i.e.* $\boldsymbol{\mu}$ and $\boldsymbol{\Phi}$) and sum over all possible topic assignments. Here, we approximate the integrals using a Gibbs sampler for each sample of assignments $\boldsymbol{z}, \boldsymbol{t}$.

In our experiments, we run each Gibbs sampler for 2,000 iterations with 1,500 burn-in. After the burn-in period, a total number of 5 samples are drawn at a lag of 100 iterations. These samples are averaged to yield the final trained model. In order to make a scientific comparison, we set hyper-parameters fairly, since they are important for the two models. Instead of using symmetric Dirichlet priors, we employ the moment-match algorithm [20] to estimate $\boldsymbol{\alpha}$ from data for LDA. For our SeqLDA model, we empirically choose parameters without optimization as: $a = 0.2$, $b = 10$, $\alpha = 0.1$. And γ is set to $200/W$ for both models. Note that we seek to optimize the parameter settings for the LDA model, which enables us to draw sound conclusions on SeqLDA's performance.

Figure 3 demonstrates the perplexity comparison for different number of topics. The LDA model has been tested on document level (LDA_D) and paragraph level (LDA_P) separately. We have also run the SeqLDA model with or without being boosted by either LDA_D (SeqLDA_D) or LDA_P (SeqLDA_P). The boosting is done by using the topic assignments learnt by the LDA model to initialize the SeqLDA model. As shown in the figure, our SeqLDA model, either with or without boosting, consistently performs better than both LDA_D and LDA_P. The p-values from the paired-t test shown in Table IV are always smaller than 0.05, which has clearly indicated that the advantage of the SeqLDA model over the LDA model is statistically significant. Evidently, the topical dependencies information propagated through the document structure, for the patent dataset, indeed exists; and explicitly considering the dependency structure in topic modelling, as our SeqLDA model does, can be valuable to help understand the original text content.

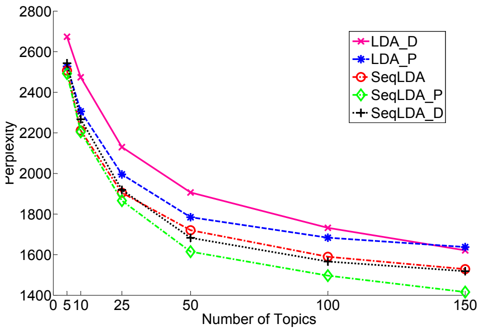


Figure 3. Perplexity comparison on the Pat-1000 dataset with 20% hold out for testing

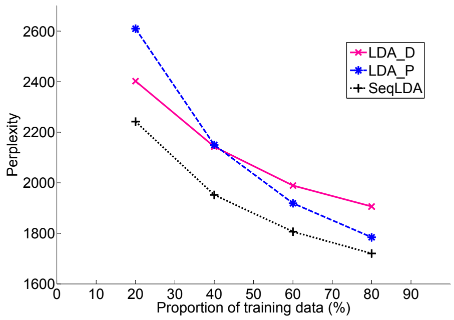


Figure 4. Perplexity comparison on the Pat-1000 dataset with different percentages of training data ($K = 50$)

In our second set of experiments, we show the perplexity comparison by changing the proportion of training data. In these experiments, the number of topics for both LDA and SeqLDA are assumed to be fixed and equal to 50. As shown in Figure 4, the SeqLDA model (without boosting) always performs better than the LDA model as the proportion of training data increases. The training time, for example, with 80% training proportion and 2000 iterations, is approximately 5 hours for LDA on document level, and 25 hours for SeqLDA.

C. Topic Distribution Profile over Segments

Besides better modelling perplexity, another key contribution of our SeqLDA model is the ability to discover underlying sequential topic evolvement within a document. With this, we can further perceive how the author organizes,

Table V
TYPICAL TOPICS LEARNT FROM "THE PRINCE". TOP 30 WORDS ARE LISTED AS EXAMPLES.

LDA	topic-0	servant servants pandolfo good opinion cares honours recognize honest comprehends venafro trust attention fails praise judgment honouring form thinking correct error clever choosing rank disposed prime useless Sinea faithfull study
	topic-9	truth emperor flatterers opinions counsels wisdom contempt advice listen preserved bold counsel resolutions speaking maximilain patient unite born deceived case affairs short anger prove receive support steadfast guarding discriminating inferred
SeqLDA	topic-0	servant flatterers pandolfo opinions truth good hones question emperor counsels form cares opinion servants wisdom comprehends enable interests honours contempt fails venafro preserved maximilain choosing advantageous listen thinking capable recognize
	topic-9	support cardinals labours fortify walls temporal fortified courageous pontificate spirits resources damage town potentates character barons burnt ecclesiastical principalities defence year firing hot attack pursuit loss showed enemy naturally
	topic-15	people nobles principality favour government times hostile ways oppressed enemies secure give messer friendly rule security courage authority satisfy arises fail rome receive finds adversity civil builds aid expect cities
	topic-16	prince men great good state princes man things make time fear considered subject found long wise army people affaires defend whilst actions life fortune difficulty present mind faithful examples roman

for instance, her/his stories in a book or her/his ideas in an essay. Here, we test SeqLDA on the two books with following parameter settings: $a = 0$, $\alpha = 0.5$, $k = 20$, $b = 25$ for "The Prince", and $b = 50$ for "The Whale".

To compare the topics of the SeqLDA and LDA models, we have to solve the problem of topic alignment, since topics learnt in separate runs have no intrinsic alignment. The approach we adopt is to start the SeqLDA's Gibbs sampling with the topic assignments learnt from the LDA model. Figures 5(a) and 6(a) show the confusion matrices between the topic distributions generated by SeqLDA and LDA with Hellinger Distance, where SeqLDA topics run along the X-axis. Most topics are well aligned (with blue on the diagonal and yellow off diagonal), especially those for "The Whale". For "The Prince", the major confusion is with topic-0 and 9 yielding some blueish off diagonal.

After aligning the topics, we plot the topic distributions (*i.e.*, sub-ideas) as a function of chapter to show how each topic evolves, shown in Figures 5 and 6 respectively. Immediately, we see that the topic evolving patterns over chapters learnt by SeqLDA are much clearer that those learnt by LDA. For example, compare two subfigures in Figure 6, it is hard to find the topic evolvement patterns in Figure 6(b) learnt by LDA; in contrast, we can find the patterns in Figure 6(c), for example, topic-7, which is about men on board ship generally, and topic-12, which is about the speech of old ("thou," "thee," "aye," "lad") co-occur together from chapters 15 to 40 and again around chapters 65-70, which is coherent with the book.

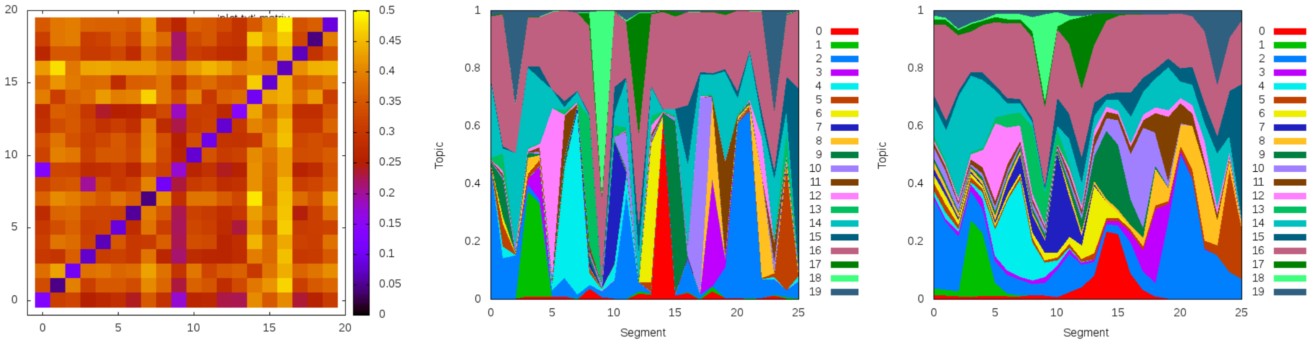
Moreover, Figures 7(a) and 7(b) depict the Hellinger distances (also as a function of chapter) between the topic distributions of two consecutive chapters (*i.e.*, between chapter i and chapter $i + 1$) to measure how smoothly topics evolve through the books. Obviously, the topic evolvement learnt by SeqLDA is much better than that learnt by LDA. SeqLDA always yields smaller Hellinger distances and smaller variance of distances. The big topic shifts found by LDA are also highlighted by SeqLDA, such as Chapter 7 to 10 in Figure 7(a). Evidently, the SeqLDA model has avoided heavy topic drifting, and makes the topic flow between chapters much smoother than LDA does. An immediate and obvious effect is that this can help the reader understand more precisely how each book is organized.

Consider "The Prince" in more detail. The topic that is most unchanged in "The Prince" is topic-16 (having the lightest yellow in off-diagonal in Figure 5(a)), also show in Table V. This topic occurs consistently through the chapters in both models and can be seen to really be the core topic of the book. Topic-15 is another topic that has not changed much, and it has its occurrence broadened considerably; for the SeqLDA model it now occurs throughout the second half of the book starting at chapter 10; the topic is about the nature of governing principalities as opposed to the first 9 chapters which cover how principalities are formed and how princes gain their title. Now consider the issue of topic-0 and 9. Inspection shows topic-9 learnt by LDA occurring in Chapters 2 and 16 is split into two by SeqLDA: the chapter 16 part joins topic-0 which has its strength in the neighbouring Chapter 15, and the topic-0 part broadens out amongst the three chapters 1-3. These topics are illustrated in Table V and it can be seen that topic-0 and topic-9 by LDA talk about related themes.

Now consider "The Whale" in more detail. In some cases SeqLDA can be seen to refine the topics and make them more coherent. Topic-6, for instance, in SeqLDA is refined to be about the business of processing the captured whale with hoists, oil, blubber and so forth. This occurs starting at chapter 98 of the book. For the LDA model this topic was also sprinkled about earlier. In other cases, SeqLDA seems to smooth out the flow of otherwise unchanged topics, as seen for topic-0, 1 and 2 at the bottom of Figure 6(c).

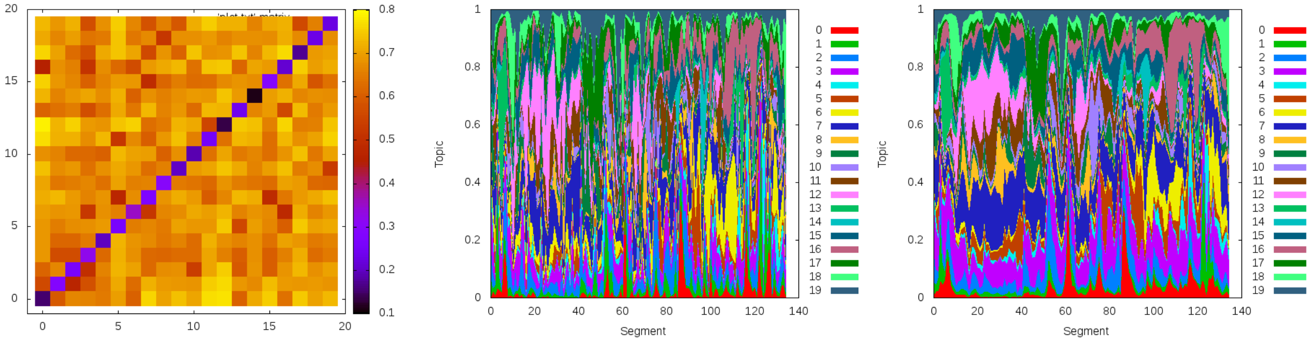
V. CONCLUSION

In this paper, we have proposed a novel generative model, the Sequential Latent Dirichlet Allocation (SeqLDA) model by explicitly considering the sequential document structure in the hierarchical modelling. We have developed for SeqLDA an efficient collapsed Gibbs sampling algorithm based on the nested two-parameter Poisson-Dirichlet process (PDP). Besides the advantage over LDA in terms of improved perplexity, the ability of the SeqLDA model to discover more coherent sequential topic structure (*i.e.*, how topics evolves among segments within a document) has been demonstrated in our experiments. The experimental results



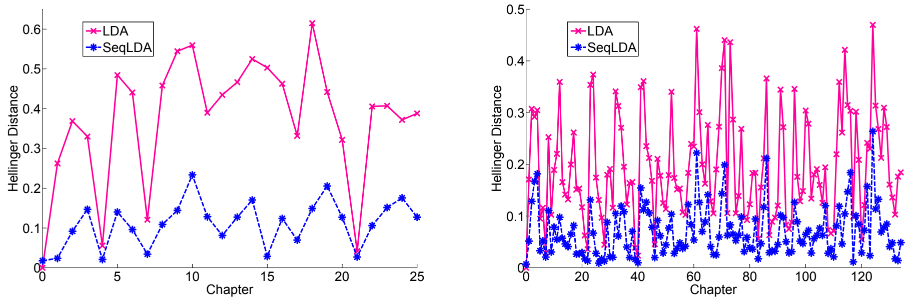
(a) Confusion Matrix (b) LDA (c) SeqLDA

Figure 5. Topic evolvement analysis for "The Prince"



(a) Confusion Matrix (b) LDA (c) SeqLDA

Figure 6. Topic evolvement analysis for "The Whale"



(a) The Prince (b) The Whale

Figure 7. Topic evolvment by Hellinger Distance

also indicate that the document structure can aids in the statistical text analysis, and structure-aware topic modelling approaches provide a solution for going beyond the "bag-of-words" assumption.

There are various ways to extend the SeqLDA model which we hope to explore in the future. The model can be applied to conduct document summarisation or document classifications, where sequential structures could play an important role. The two parameters a and b in the PDP can be optimized dynamically for each segment, instead of fixed, in order to handle the topic drifting problem for few segments *i.e.* when the correlations between two successive segments are not very strong.

Acknowledgment

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program. Dr. Huidong Jin for this work was partly supported by the CSIRO-Centrelink Human Services Delivery Research Alliance.

References

[1] W. Buntine and A. Jakulin, "Discrete components analysis," in *Subspace, Latent Structure and Feature Selection Techniques*. Springer-Verlag, 2006.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[3] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proc. of the 23rd ICML*, 2006, pp. 113–120.

[4] X. Wang and A. McCallum, "Topics over time: a non-Markov continuous-time model of topical trends," in *Proc. of the 12th SIGKDD*, 2006, pp. 424–433.

[5] L. AlSumait, D. Barbará, and C. Domeniconi, "On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking," in *Proc. of the 8th ICDM*, 2008, pp. 3–12.

[6] R. M. Nallapati, S. Ditmore, J. D. Lafferty, and K. Ung, "Multiscale topic tomography," in *Proc. of the 13th SIGKDD*, 2007, pp. 520–529.

[7] T. L. Griffiths, M. Steyvers, D. M. Blei, and J. B. Tenenbaum, "Integrating topics and syntax," in *NIPS 17*, 2005, pp. 537–544.

[8] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *Proc. of the 20th UAI*, 2004, pp. 487–494.

[9] H. Wang, M. Huang, and X. Zhu, "A generative probabilistic model for multi-label classification," in *Proc. of the 8th ICDM*, 2008, pp. 628–637.

[10] D. Blei and J. Lafferty, "Correlated topic models," in *NIPS 18*, 2006, pp. 147–154.

[11] D. M. Blei, T. L. Griffiths, and M. I. Jordan, "The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies," *J. ACM*, vol. 57, no. 2, pp. 1–30, 2010.

[12] M. M. Shafiei and E. E. Milios, "Latent Dirichlet coclustering," in *Proc. of the 6th ICDM*, 2006, pp. 542–551.

[13] J. Pitman and M. Yor, "The two-parameter Poisson-Diriclet distribution derived from a stable subordinator," *Annals of Probability*, vol. 25, no. 2, pp. 855–900, 1997.

[14] H. Ishwaran and L. F. James, "Gibbs sampling methods for stick breaking priors," *Journal of the American Statistical Association*, pp. 161–173, March 2001.

[15] W. Buntine and M. Hutter, "A Bayesian review of the Poisson-Dirichlet process," *Submitted for publication*, 2010.

[16] Y. Teh, "A Bayesian interpretation of interpolated Kneser-Ney," School of Computing, National University of Singapore, Tech. Rep. TRA2/06, 2006.

[17] Q. He, B. Chen, J. Pei, B. Qiu, P. Mitra, and L. Giles, "Detecting topic evolution in scientific literature: how can citations help?" in *The 18th CIKM*, 2009, pp. 957–966.

[18] T. L. Griffiths and M. Steyvers, "Finding scientific topics." *Natl Acad Sci USA*, vol. 101 Suppl 1, pp. 5228–5235, 2004.

[19] L. Du, W. Buntine, and H. Jin, "A segmented topic model based on the two-parameter Poisson-Dirichlet process," *Mach. Learn.*, vol. 81, no. 1, pp. 5–19, 2010.

[20] T. P. Minka, "Estimating a Dirichlet distribution," MIT, Tech. Rep., 2000.

Appendix

Two-parameter Poisson Dirichlet process and Chinese Restaurants

The two-parameter Poisson-Dirichlet process (PDP), is a generalization of the Dirichlet process. In regard to SeqLDA, let $\boldsymbol{\mu}$ be a distribution over topics (*i.e.* topic proportion). We recursively place a PDP prior on $\boldsymbol{\mu}_j$ $(j \geq 1)$: $\boldsymbol{\mu}_j \sim \text{PDP}(a, b, \boldsymbol{\mu}_{j-1})$, where the three parameters are: a base distribution $\boldsymbol{\mu}_{j-1}$; a $(0 \leq a < 1)$ and b $(b > -a)$. The parameters a and b can be understood as controlling the amount of variability around the based distribution [16].

Here, we give a brief discussion of the PDP within the Chinese restaurant process model. Consider a sequence of N customers sitting down in a Chinese restaurant with an infinite number of tables each with infinite capacity but each serving a single dish. Customers in the CRP are words in our model, and dishes in the CRP are topics. The basic process with $\boldsymbol{\mu}$ marginalized out is specified as follows: the first customer sits at the first table; the $(n+1)^{th}$ subsequent customer sits at the t^{th} table (for $1 \leq t \leq T$) with probability $\frac{n_t^* - a}{b+n}$, or sits at the next empty ($(T+1)^{th}$) table with probability $\frac{b + T \times a}{b+n}$. Here, T is the current number of occupied tables in the restaurant, and n_t^* is the number of customers currently sitting at table t. The customer takes the dish assigned to that table, for table t given by k_t^*. Therefore, the posterior distribution of the $(n+1)^{th}$ customer's dish is

$$\frac{b + T \times a}{b+n} \boldsymbol{\mu} + \sum_{t=1}^{T} \frac{n_t^* - a}{b+n} \delta_{k_t^*}(\cdot)$$

where k_t^* indicates the distinct dish associated with the t^{th} table, and $\delta_{k_t^*}(\cdot)$ places probability one on the outcome k_t^*.

In general PDP theory, the dishes (or values) at each table can be any measurable quantity, but in our case they are a finite topic index $k \in \{1, \cdots, K\}$. This finite discrete case has some attractive properties shown in [15], which follows some earlier work of [16]. To consider this case we introduce another latent constraint variable: t_k, the *table count* of menu k. In this discrete case, given a probability vector $\boldsymbol{\mu}$ of dimension K, and the following set of priors and likelihoods for $j = 1, ..., J$

$$\begin{aligned} \boldsymbol{\mu}_j &\sim \text{PDP}(a, b, \boldsymbol{\mu}_{j-1}) \\ \boldsymbol{m}_j &\sim \text{multinomial}_K(\boldsymbol{\mu}_{j-1}, M_j) \end{aligned}$$

where $M_j = \sum_k m_{j,k}$. Introduce auxilliary latent variables $\boldsymbol{t}_j$ such that $t_{j,k} \leq m_{j,k}$ and $t_{j,k} = 0$ if and only if $m_{j,k} = 0$, then the following marginalised posterior distribution holds

$$\begin{aligned} & p(\boldsymbol{n}_j, \boldsymbol{t}_j | a, b, \boldsymbol{\mu}_{j-1}) \\ & \quad = C_{\boldsymbol{n}_j}^{M_j} \frac{(b|a)_{\sum_k t_{j,k}}}{(b)_{M_j}} \prod_k S_{t_{j,k},a}^{m_{j,k}} \prod_k \mu_{j-1,k}^{t_{j,k}} \,. \qquad (6) \end{aligned}$$

where $C_{\boldsymbol{n}_j}^{M_j}$ is the multi-dimensional choose function of a multinomial. Note that in the nested PDP we consider in the SeqLDA model, a table in any given restaurant reappears as a customer in its parent restaurant due to the last product term in Equation (6). Thus, there are two types of customers in each restaurant using the notation of Table II, the ones arriving by themselves ($\boldsymbol{n}_j$), and those sent by its child restaurant ($\boldsymbol{t}_{j+1,k}$).