Knowl Inf Syst (2012) 31:475–503
DOI 10.1007/s10115-011-0425-1

REGULAR PAPER

Sequential latent Dirichlet allocation

Lan Du · Wray Buntine · Huidong Jin · Changyou Chen

Received: 8 January 2011 / Revised: 14 April 2011 / Accepted: 24 May 2011 /
Published online: 10 June 2011

Abstract Understanding how topics within a document evolve over the structure of the document is an interesting and potentially important problem in exploratory and predictive text analytics. In this article, 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 antecedent and subsequent segments. Such progressive sequential dependency is captured by using the hierarchical two-parameter Poisson–Dirichlet process (HPDP). We develop an efficient collapsed Gibbs sampling algorithm to sample from the posterior of the SeqLDA based on the HPDP. Our experimental results on patent documents show that by considering 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 several books such as Melville's 'Moby Dick'.

Keywords Latent Dirichlet allocation · Poisson–Dirichlet process · Collapsed Gibbs sampler · Topic model · Document structure

L. Du · W. Buntine · H. Jin · C. Chen
CECS, The Australian National University, Canberra, ACT, Australia

L. Du (✉) · W. Buntine · C. Chen
National ICT Australia, Building A, 7 London Circuit, Canberra, ACT 2601, Australia
e-mail: Lan.Du@nicta.com.au

W. Buntine
e-mail: Wray.Buntine@nicta.com.au

C. Chen
e-mail: Changyou.Chen@nicta.com.au

H. Jin
CSIRO Mathematics, Informatics and Statistics, Canberra, ACT, Australia
e-mail: Warren.Jin@csiro.au

1 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 [6,10]. Its fundamental idea is that documents can be taken as mixtures of latent topics, each of which is a probability distribution over words in a vocabulary, under the '*bag-of-words*' assumption. The topic distributions provide an explicit representation of the subject matter of a document. Indeed, the probabilistic procedures specified by topic models, as shown in Fig. 2, describe a way by which words in documents can be generated on the basis of these latent topics. Those procedures can be inverted with standard statistical techniques to infer the latent topics. Although topic modelling is a probabilistic generative process, it can also be viewed as non-negative matrix factorisation (NMF) [20,25,35], finding a non-negative matrix product close in Kullback–Leibler divergence. In this sense, topic models break the document–word frequency matrix into a topic–word matrix and a document–topic matrix.

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 analyse the content of documents and the meaning of words. These include models of words only [6,17], of topic trends over time [2,5,23,38], of word-order with Markov dependencies [15], of words and supervised information [7,21], e.g. authors [29], class labels [37], of the intra-topic correlation (i.e. the hierarchical structure of topics) [3,4], of segments in documents [30], 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.

It is known that 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 big challenge in text mining is the problem of understanding the subject structure of a document, and of further incorporating this structure into the analysis of the original text.

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 organised, accessible and understandable to readers. As we read and interpret documents, we should bear in mind correlations between main idea and sub-ideas (which have been studied by the segmented topic model (STM) [11]), and correlations between sub-ideas of adjacent segments. Apparently, a good document structure, as exemplified in Fig. 1, should have the aforementioned features, which can make, for instance, the arguments of an essay cohesive and flow logically. Therefore, we believe segments not only have meaningful content but also provide contextual information for subsequent segments.

Can we statistically analyse 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, and simply neglect the contextual information buried in the document structure. Although the latent Dirichlet co-clustering (LDCC) Model [30], as shown in Fig. 2b,

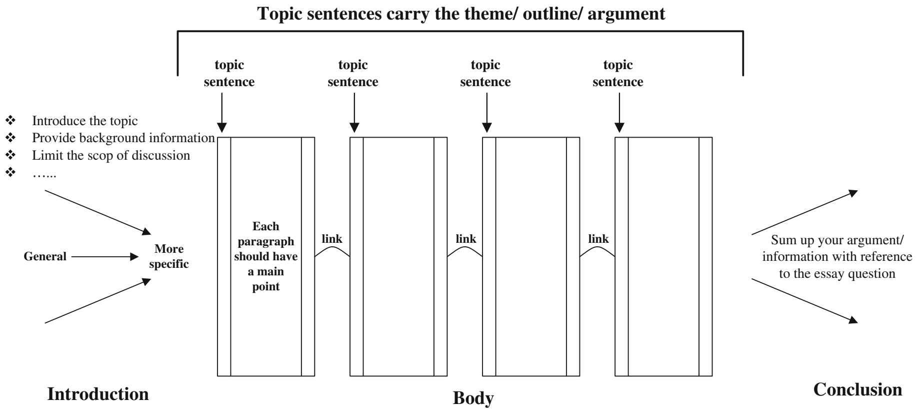
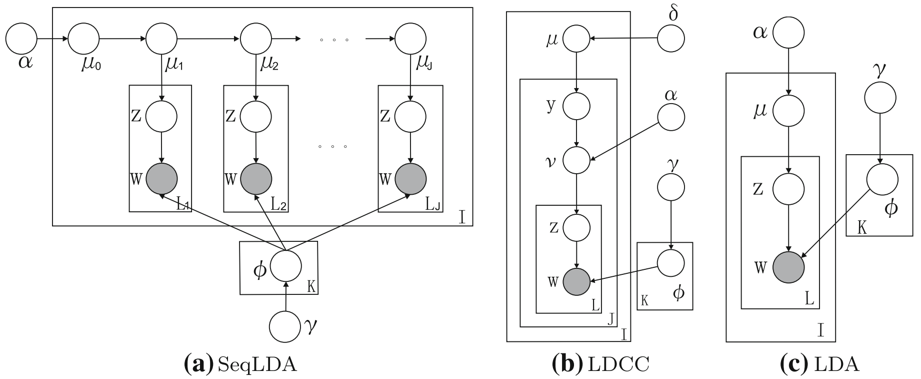

Topic sentences carry the theme/ outline/ argument
topic sentence
topic sentence
topic sentence
topic sentence
❖ Introduce the topic
❖ Provide background information
❖ Limit the scop of discussion
❖ ……
General
More specific
Each paragraph should have a main point
link
link
link
Sum up your argument/ information with reference to the essay question
Introduction
Body
Conclusion

Fig. 1 An example document structure: an essay structure



(a) SeqLDA **(b)** LDCC **(c)** LDA

Fig. 2 Graphical model representations for the SeqLDA model, the LDCC model and the LDA model

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. [15] proposed a model that captures both syntactic (i.e. word-order) and semantic (i.e. topic) dependencies, by using Markov dependencies and topic models, respectively. But, topical dependency buried in the higher levels of document structure, such as amongst paragraphs, is not modelled.

Different from previous topic models, this article presents a new variant of the latent Dirichlet allocation (LDA) model [6], a topic model, called sequential latent Dirichlet allocation (SeqLDA), that explicitly models the underlying document structure. In this work, we restrict ourselves to the study of the sequential topic structure of a document, that is how a sub-idea in a segment is closely related to its antecedent and subsequent segments. In our SeqLDA 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 preceding segment, the one for the first segment will depend on the initial topic distribution (i.e. the document topic distribution). The progressive topical dependency is captured using the two-parameter Poisson–Dirichlet process (PDP) [18,26] in a hierarchical way, based on theoretical results in the finite discrete space [8,9,11].

The PDP is also known as the Pitman–Yor Process, and related to the Chinese restaurant process (CRP). The PDP is here denoted as $\boldsymbol{u} \sim \text{PDP}(a, b, \boldsymbol{v})$, where a is called the discount parameter, b the strength parameter and $\boldsymbol{v}$ a discrete base distribution (whilst the PDP applies to general distributions, only discrete distributions are considered here). In our case, the base distribution $\boldsymbol{v}$ is to be a probability vector and the hierarchical PDP is used to create a distribution over a hierarchy of probability vectors. From an introductory perspective, the PDP can be understood as being approximately like the Dirichlet, so $\boldsymbol{u} \sim \text{PDP}(a, b, \boldsymbol{v})$ is somewhat like $\boldsymbol{u} \sim \text{Dirichlet}(\frac{1-a}{1+b}\boldsymbol{v})$ [9]. This means $\boldsymbol{u}$ has a mean of $\boldsymbol{v}$ and a variance proportional to $\frac{1+b}{1-a}$.

Our reason for using the Pitman–Yor Process instead of the Dirichlet is pragmatic. When we sample from $\boldsymbol{u} \sim \text{PDP}(a, b, \boldsymbol{v})$, especially where we have a network of these probability vectors, the Dirichlet would yield an intractable posterior whereas the PDP allows tractable Gibbs sampling with auxiliary variables. The Dirichlet posterior contains terms like $u_i^{\alpha v_i}$ (for some constant α), which then combined with some terms from a hierarchy like $v_i^{m_i}$ yield an intractable posterior. In contrast, the PDP posterior has terms like $v_i^{t_i}$, where t_i is an auxiliary variable, a latent integer count, usually much less than the data count. The PDP posterior introduces latent counts $\boldsymbol{t}$ called table counts corresponding to the data counts $\boldsymbol{n}$ occurring in the form of $v_i^{t_i}$, which is conjugate to a multinomial with parameter $\boldsymbol{v}$. When combined with some terms from a hierarchy like $v_i^{m_i}$, the probability vector $\boldsymbol{v}$ can even be integrated out of the posterior [9,11,32].

The hierarchical PDP (HPDP) is defined on a singly connected network of probability vectors $\boldsymbol{u}_i$ for nodes i. In our case, the network is a chain and $\boldsymbol{u}_{i-1}$ is the parent of $\boldsymbol{u}_i$. The HPDP is defined by placing PDP distributions on the network so $\boldsymbol{u}_i \sim \text{PDP}(a_i, b_i, \boldsymbol{u}_{i-1})$, where a_i is discount parameter, b_i is strength parameter and $\boldsymbol{u}_{i-1}$ is base distribution for $\boldsymbol{u}_i$, as done in [32,33]. As just explained, the advantage of using the HPDP is that it allows us to integrate out the real valued probability vectors $\boldsymbol{u}_i$, i.e. the PDP is conjugate to itself when applied to the discrete data. In this article, we use the term *self-conjugate* to refer to this property. We also develop here a collapsed Gibbs sampling algorithm for the HPDP in its linear form used here.

Using the HPDP chain of probability vectors let us model sequential structure. We can explore how topics are evolving amongst, 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. Tackling topic modelling together with the topical structure buried in a document provides a solution for going beyond the bag-of-words assumption, which is widely used in text analytics (e.g. natural language processing and information retrieval).

The rest of the article, which extends an earlier contribution [12], is organised as follows. We first briefly discuss the related work in Sect. 2. Then, we describe the SeqLDA model in detail, and compare it with some related models (e.g. LDA, LDCC and STM) in Sect. 3. Section 4 elaborates an efficient collapsed Gibbs sampling algorithm based on the PDP for our SeqLDA. We present in Sect. 5, qualitative results that demonstrate how the SeqLDA allows the exploration of a document in a new way, and quantitative results that demonstrate greater predictive accuracy when compared with the LDA model. Section 6 gives a brief discussion and concluding comments.

2 Related work

To capture topic evolvement in temporal data, the integration of timestamps into topic models has been around for a while. Existing work focuses mainly on learning topic evolvement patterns from a time-varying corpus, instead of exploring how topics progress within each individual document by following the latent document structure. These works explore how topics change, rise and fall, by considering timestamps associated with a corpus. In general, they can be put into two categories, Markov chain-based models and non-Markov chain-based models.

In the Markov chain-based models, the dynamic behaviours (i.e. topic evolvement in our perspective) are captured by state transitions. The state at time $t + \Delta_t$ is dependent on the state of t. For instance, the dynamic topic model (DTM) [5], the dynamic mixture models (DMM) [39], the dynamic hierarchical Dirichlet process [1,28], the evolutionary Hierarchical Dirichlet Process (EvoHDP) [40] and so on.

The DTM captures the topic evolution in a document collection that is organised sequentially into several discrete time periods, and then within each period an LDA model is trained on its documents. The Gaussian distributions are used to tie a collection of LDAs by chaining the Dirichlet prior and the natural parameters of each topic. However, Gaussian distributions are not conjugate to multinomial distributions. Wang et al. [36] extend DTM to a continuous time space to resolve the problem of discretisation by adopting the Brownian motion model. The DMM assumes that the mixture of latent variables (i.e. topic distribution) for all streams is dependent on the mixture of the previous timetamp. We note that even though the structure of the DMM is similar to the SeqLDA (i.e. both put first-order Markov assumptions on topic distributions), our model capitalises on the self-conjugacy of the PDP to chain a series of LDAs, instead of using Dirichlet distributions. The problem with the Dirichlet distribution is that it is not self-conjugate, which could not facilitate an efficient inference algorithm.

Recently, the hierarchical Dirichlet process (HDP) [34] has been extended to incorporate time dependence to model the time-evolving properties of sequential data sets. The dynamic HDP model (DHDP) [28] ties a series of HDP by using Dirichlet processes (DP). The DP is used to generate innovation distributions at different timestamps. Then, at each timestamp, the corresponding innovation distribution is mixed with the distribution at its previous timestamp to yield the target distribution. Since the DHDP only models the evolutionary patterns in a single dynamic corpus, Zhang et al. [40] propose the EvoHDP model by extending the HDP to handle multiple correlated time-varying corpora. They put the Markov assumption on both global and local measures, which results in one more layer than the DHDP. Furthermore, the infinite dynamic topic model (IDTM) proposed in [1] is another dynamic version of HDP, which can handle the birth/death of topics, as declared by the authors.

The other type of models do not assume the Markovian dependence over time, but instead treat time as an observed variable that can be jointly generated with words by the latent topics, for example, the topics over time (ToT) model [38]. In the ToT, the topic over time is captured by a beta distribution. Drawing all time stamps from the same beta distribution might be not appropriate for, such as, stream data [39]. Some other approaches are, for instance, He et al. [16] develop inheritance topic model to understand topic evolution by leveraging the citation information; Kandylas et al. [19] analyse the evolution of knowledge communities based on the clustering over time method, called Streemer.

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 capitalising on the correlations amongst its segments, i.e. the underlying sequential topic structure, according to the original document layout. The Markov dependencies are put on the topic distributions. In this way, we can directly model the topical dependency between a segment and its successor.

Although one may argue that the models we just discussed can also be applied to the individual document by treating the sequence of segments as timestamps, the computation complexity and space complexity of those models can be significantly increased with the growth of the latent variables and hyper-parameters. In contrast, we use a single integrated model based on the HPDP, in which the real valued parameters can be integrated out because of self-conjugacy.

Note that collapsed Gibbs sampling has been widely adopted in topic modelling [2,4,29–31,40], since it was first proposed by Griffiths and Steyvers in 2004 [14]. It can give us a relatively simple algorithm for approximate inference, because some latent variables can be integrated out due to the conjugacy. Recently, it has been improved to handle the challenge caused by the scalability of various data sets, such as a distributed Gibbs sampling algorithm [24] and fast Gibbs sampling algorithm [27].

In regard to the PDP, the existing Gibbs sampling algorithms are built on top of either the CRP [4,32–34] or the stick breaking construction [18,28,40]. Here, we consider those based on the CRP. The standard implementation of Gibbs sampling for the PDP needs to record the full configuration of the customer seating plan, which is termed 'sampling for seating arrangement' by Teh [32]. In contrast, our Gibbs sampling algorithm has summed out all the seating arrangement by introducing a new auxiliary latent variable, similar to that proposed in [11]. It can further facilitate the development of a hierarchical version. Details will be discussed later in this article.

3 Sequential latent Dirichlet allocation

In this section, we present the novel Sequential Latent Dirichlet Allocation model (SeqLDA) which models how topics evolve amongst segments in a document. We assume that there could be some latent sequential topic structures within each individual document, i.e. the ideas within a document evolve smoothly from one segment to another, especially in various books (e.g. novels and textbooks). This assumption intuitively originates from the way in which people normally organise ideas in their writing. Before specifying the SeqLDA model, we list notation and terminology used in this article. Notation is depicted in Table 1. We define the following terms and dimensions:

- A *word* is the basic unit of our data, selected from a vocabulary indexed by $\{1, \ldots, W\}$.
- A *segment* is a group of L words. It can be a chapter, section, paragraph or sentence. In this work, we assume segments are either paragraphs or chapters.
- 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. Obviously, both the document and its segments share the same topic space. 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. Take the book, called 'The Prince', as a example. The whole book is treated as a document, each chapter is a segment in our experiments, refer to Sect. 5.3. The subject of each chapter is simulated by the distribution (i.e. $\mu_{i,j}$) over latent topics. The linkage between subjects is modelled by the change between topic distributions.

Table 1 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 document i and segment j
$\boldsymbol{\Phi}$	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

The development of a sequential structural generative model according to the above idea is based on the HPDP, and models how the sub-idea of a segment is correlated to its previous and following segment. Specifically, the correlation is simulated by the progressive dependency amongst topic distributions. That is, the jth 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_i and discount parameter a_i control the variation between the adjacent topic distributions. Figure 2a 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 describes 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, …, K}
   (a) Draw $\boldsymbol{\phi}_k \sim \text{Dirichlet}_W(\boldsymbol{\gamma})$
2. For each document $i$
   (a) Draw $\boldsymbol{\mu}_{i,0} \sim \text{Dirichlet}_K(\boldsymbol{\alpha})$
   (b) For each segment $j \in \{1, \ldots, J_i\}$
       i  Draw $\boldsymbol{\mu}_{i,j} \sim \text{PDP}(a_i, b_i, \boldsymbol{\mu}_{i,j-1})$
       ii For each word $w_{i,j,l}$, where $l \in \{1, \ldots, L_{i,j}\}$
          A draw $z_{i,j,l} \sim \text{multinomial}_K(\boldsymbol{\mu}_{i,j})$
          B draw $w_{i,j,l} \sim \text{multinomial}_W(\boldsymbol{\phi}_{z_{i,j,l}})$
```

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 $\boldsymbol{\Phi} = (\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_K)$, and will be estimated through 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, but the PDP is also self-conjugate. 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 Fig. 2a using the distributions given in the above generative process, as below:

$$p(\boldsymbol{\mu}_{i,0}, \boldsymbol{\mu}_{i,1:J}, \boldsymbol{z}, \boldsymbol{w}|\boldsymbol{\alpha}, \boldsymbol{\Phi}, a_i, b_i) \\ = p(\boldsymbol{\mu}_{i,0}|\boldsymbol{\alpha}) \prod_{j=1}^{J_i} p(\boldsymbol{\mu}_{i,j}|a_i, b_i, \boldsymbol{\mu}_{i,j-1}) \prod_{l=1}^{L_j} p(z_{i,j,l}|\boldsymbol{\mu}_{i,j}) p(w_{i,j,l}|\boldsymbol{\phi}_{z_{i,j,l}}) \tag{1}$$

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

From the notion of the proposed model, we can find the obvious distinction between the SeqLDA model and the LDA model (shown in Fig. 2c): the SeqLDA model takes into account the sequential structure of each document, i.e. the position of each segment that the LDA model ignores. Our SeqLDA model aims to capitalise 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 [30], as shown in Fig. 2b, still has an implicit assumption that segments within a document are exchangeable, not always appropriate, so does STM [11]. Furthermore, assigning just one topic to each segment in the LDCC cannot capture the evolvement of each topic depicted in the document. Like the SeqLDA model, the STM assumes each segment has a topic distribution, and each segment topic distribution is drawn from document topic distribution by the PDP. As discussed earlier, the STM is developed to explore only the relationship between a main idea and its corresponding sub-ideas. The exchangeability assumption posed by the STM may make it unsuitable for describing the sequential topic structure and detecting the topic evolvement.

Thus, if documents indeed have some latent sequential structure, considering this dependency means a higher fidelity of SeqLDA over LDA and LDCC. However, if the correlation amongst sub-ideas of segments is not obvious, taking the topic distribution of the jth segment as the base distribution of the $(j+1)$th segment may misinterpret 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 optimising strength and discount parameters (a and b) for each individual segment within each document. Though for simplicity, we first fix a and b for each corpus, and then optimise b_i for each document i with a fixed in all our reported experiments.

4 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 $\boldsymbol{\mu}_{0:J}$ and topic assignment $\boldsymbol{z}$) given the inputs (i.e. $\boldsymbol{\alpha}$, $\boldsymbol{\Phi}$, a and b) and observations $\boldsymbol{w}$, that is:

$$p(\boldsymbol{\mu}_{0:J}, \boldsymbol{z}|\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\Phi}, a, b) = \frac{p(\boldsymbol{\mu}_{0:J}, \boldsymbol{z}, \boldsymbol{w}|\boldsymbol{\alpha}, \boldsymbol{\Phi}, a, b)}{p(\boldsymbol{w}|\boldsymbol{\alpha}, \boldsymbol{\Phi}, a, b)}.$$

Table 2 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 $n_{i,j,k} = \sum_l 1_{z_{i,j,l}=k}$
$N_{i,j}$	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 kth value. Necessarily, $t_{i,j,k} \leq n_{i,j,k}$ and $t_{i,j,k} > 0$ whenever $t_{i,j,k} > 0$. In particular, if $n_{i,j,k} = 1$ then $t_{i,j,k} = 1$
$T_{i,j}$	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., $(t_{i,j,1}, \ldots, t_{i,j,K})$ for segment j
$u_{i,k}$	The smallest segment index j' in i, where $t_{i,j',k} = 0$

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. $\boldsymbol{\mu}_{0:J}$ and $\boldsymbol{\Phi}$ in our case) can be marginalised out, rather than explicitly estimated. In topic modelling literature, two standard approximation methods have often been used: variational inference [6] and Gibbs sampling [14]. Here, we pursue an alternative approximating strategy using the latter by taking advantage of the collapsed Gibbs sampler for the PDP [9,11].

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 capitalises 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. Note that we use conjugate priors in our model, i.e. Dirichlet prior $\boldsymbol{\alpha}$ on $\boldsymbol{\mu}_0$ and $\boldsymbol{\gamma}$ on $\boldsymbol{\Phi}$, PDP prior on $\boldsymbol{\mu}_j$ (PDP is self-conjugate); thus $\boldsymbol{\mu}_{0:J}$ and $\boldsymbol{\Phi}$ can be integrated out. Although the proposed sampling algorithm does not directly estimate $\boldsymbol{\mu}_{0:J}$ and $\boldsymbol{\Phi}$, we will show how they can be approximated using the posterior sample statistics.

In this section, we derive the collapsed Gibbs sampling algorithm for doing inference, and parameter estimation in the proposed model. Table 2 lists all the statistics required in our algorithm. Our SeqLDA sampling is a collapsed version of what is known as the nested Chinese restaurant process (CRP) used as a component of different topic models [4]. The basic theory of the CRP and our collapsed version of it are summarised in the '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 $k = 1, \ldots, K$, 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 (shown in Fig. 3) 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 [9,32] 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 $t_{i,j,k}$.

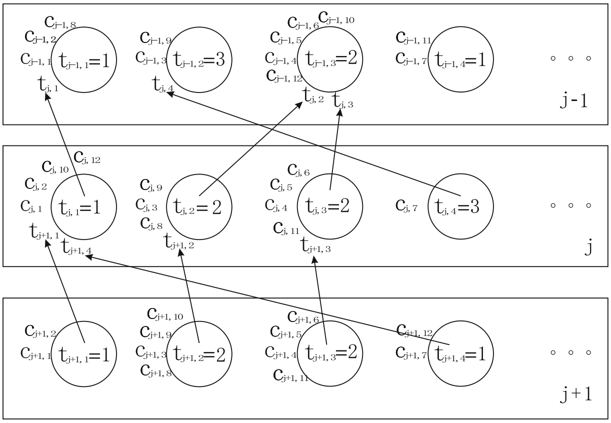


Fig. 3 The Chinese restaurant construction of the SeqLDA. Each restaurant is represented by a rectangle. Customers ($c_{j,n}$'s) are seated at tables (circles) in the restaurants. At each table a dish is served as indicated by $t_{j,m} = k$, which means the kth dish is served at the mth table in restaurant j. Note that in the hierarchical Chinese restaurant there are two types of customers, $c_{i,n}$ who arrive by themselves, and $t_{j,m}$ who are sent by the child restaurant

4.1 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 $\boldsymbol{\mu}_{0:J}$ and $\boldsymbol{\Phi}$, since they can be interpreted as statistics of the associations amongst $\boldsymbol{w}$, $\boldsymbol{z}$ and $\boldsymbol{t}$. Hence, we first recursively apply the collapsed Gibbs sampling function for the PDP, i.e., Eq. (13) in the 'Appendix A', to integrating out the segment topic distributions $\boldsymbol{\mu}_{i,1:J}$ from Eq. (1), we derive $p(\boldsymbol{z}_{1:I}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I}, \boldsymbol{\mu}_{1:I,0} \,|\, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\Phi}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I})$

$$\prod_i \frac{1}{\mathrm{Beta}_K(\boldsymbol{\alpha})} \prod_k \mu_{i,0,k}^{\alpha_k+t_{i,1,k}-1} \prod_j \frac{(b_i|a_i)_{T_{i,j}}}{(b_i)_{N_{i,j}+T_{i,j+1}}} \prod_{j,k} S^{n_{i,j,k}+t_{i,j+1,k}}_{t_{i,j,k},a_i} \prod_{w,k} \phi_{k,w}^{M_{i,k,w}} \tag{2}$$

where $t_{i,j,k} \le n_{i,j,k} + t_{i,j+1,k}$ and $t_{i,j,k} = 0$ *iff* $n_{i,j,k} + t_{i,j+1,k} = 0$; $\mathrm{Beta}_K(\boldsymbol{\alpha})$ is a K dimensional beta function that normalises 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

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

where $\Gamma(\cdot)$ denotes the standard gamma function; $S^N_{M,a}$ is a generalised Stirling number given by the linear recursion [9,32]

$$S^{N+1}_{M,a} = S^N_{M-1,a} + (N - Ma) S^N_{M,a},$$

for $M \le N$. 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.

Figure 3 shows how the segment level topic distributions can be marginalised out in a recursive way to yield Eq. (2), especially the middle three products. The arrows indicated the tables in the child restaurant are sent to its parent restaurant as proxy customers, so the total number of customers in restaurant j (or segment j) now is $N_{i,j} + T_{i,j+1}$. This is how the third and fourth products in the middle of Eq. (2) can be derived with Eq. (13).

Finally, 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. we can compute the joint conditional distribution of $\boldsymbol{z}_{1:I}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i}$ as

$$p(\boldsymbol{z}_{1:I}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I} \,|\, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I}) \\ = \prod_i \frac{\mathrm{Beta}_K\left(\boldsymbol{\alpha} + \boldsymbol{t}_{i,1}\right)}{\mathrm{Beta}_K\left(\boldsymbol{\alpha}\right)} \prod_{i,j} \frac{(b_i|a_i)_{T_{i,j}}}{(b_i)_{N_{i,j}+T_{i,j+1}}} \prod_{i,j,k} S^{n_{i,j,k}+t_{i,j+1,k}}_{t_{i,j,k},a_i} \prod_k \frac{\mathrm{Beta}_W\left(\boldsymbol{\gamma} + \boldsymbol{M}_k\right)}{\mathrm{Beta}_W\left(\boldsymbol{\gamma}\right)} \quad (3)$$

The derivations are provided in detail in 'Appendix B'.

4.2 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. Note that sampling table counts from the latter can be taken as a stochastic process of rearranging the seating plan of a Chinese restaurant in the CRP representation of the PDP.

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

where $z_{i,j,l} = k$ indicates the assignment of the lth word in the jth segment of document i to topic k, $\boldsymbol{z}_{1:I} - \{z_{i,j,l}\}$ presents all the topic assignments not including the lth word, and $\boldsymbol{t}_{1:I,1:J_i} - \{t_{i,j,k}\}$ denotes all the table counts except for the current table count $t_{i,j,k}$. Before elaborating the derivation of these two distributions, we discuss the aforementioned constraints on the table count ($t_{i,j,k}$) and the word count ($n_{i,j,k}$) for each topic. 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 Eq. (3) and also seen from Eq. (13) in the 'Appendix A', tables of $(j+1)$th restaurant are customers of jth restaurant in nested CRPs, as depicted in Fig. 3. 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 occupied 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. Handling the constraints on all table counts $t_{i,j,k}$ is the key challenge in the development of our Gibbs algorithm.

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 jth 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 $t_{i,j,k}$ go 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. (3), using the chain rule, and taking into account all cases, we obtain the final full conditional distribution

$$p(z_{i,j,l} = k \mid \boldsymbol{z}_{1:I} - \{z_{i,j,l}\}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i}, \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I}) \\ = \frac{p(\boldsymbol{z}_{1:I}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i} \mid \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I})}{p(\boldsymbol{z}_{1:I} - \{z_{i,j,l}\}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i} \mid \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I})}$$

with three different cases according to the value of $u_{i,k}$ as follows: when $u_{i,k} = 1$, which means all the table counts $t_{i,j',k}$ for $1 \leq j' \leq J_i$ are zero, we have

$$p(z_{i,j,l} = k \mid \boldsymbol{z}_{1:I} - \{z_{i,j,l}\}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i}, \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I}) \\ = \frac{\left(\alpha_k + t'_{i,1,k}\right)\left(b_i + a_i T'_{i,1}\right)}{\sum_k \alpha_k + \sum_k t'_{i,1,k}} \prod_{j'=2}^{j} \left(\frac{b_i + a_i T'_{i,j'}}{b_i + N_{i,j'-1} + T'_{i,j'}}\right) \frac{\gamma_{w_{i,j,l}} + M'_{k,w_{i,j,l}}}{\sum_w (\gamma_w + M'_{k,w})} \qquad (4)$$

When $1 < u_{i,k} \leq j$, which means all the table counts $t_{i,j',k}$ for $u_{i,k} \leq j' \leq J_i$ are zero, the conditional probability is

$$p(z_{i,j,l} = k \mid \boldsymbol{z}_{1:I} - \{z_{i,j,l}\}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i}, \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I}) \\ = \prod_{j'=u_{i,k}}^{j} \left(\frac{b_i + a_i T'_{i,j'}}{b_i + N_{i,j'-1} + T'_{i,j'}}\right) \frac{S^{n_{i,u_{i,k}-1,k}+1}_{t_{i,u_{i,k}-1,k},a_i}}{S^{n_{i,u_{i,k}-1,k}}_{t_{i,u_{i,k}-1,k},a_i}} \frac{\gamma_{w_{i,j,l}} + M'_{k,w_{i,j,l}}}{\sum_w (\gamma_w + M'_{k,w})} \qquad (5)$$

When $j < u_{i,k}$, which means the current table count $t_{i,j,k} > 0$ (no recursive check), it is simplified to

$$p(z_{i,j,l} = k \mid \boldsymbol{z}_{1:I} - \{z_{i,j,l}\}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i}, \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I}) \\ = \frac{S^{n'_{i,j,k}+1+t_{i,j+1,k}}_{t'_{i,j,k},a_i}}{S^{n'_{i,j,k}+t_{i,j+1,k}}_{t'_{i,j,k},a_i}} \frac{\gamma_{w_{i,j,l}} + M'_{k,w_{i,j,l}}}{\sum_w (\gamma_w + M'_{k,w})} \tag{6}$$

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, 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:

$$t_{i,j,k} \in \left[\max\left(1, t_{i,j-1,k} - n_{i,j-1,k}\right), n_{i,j,k} + t_{i,j+1,k} \right]$$

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.

$$p(t_{i,j,k} \mid \boldsymbol{z}_{1:I}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i} - \{t_{i,j,k}\}, \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I}) \\ = \frac{p(\boldsymbol{z}_{1:I}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i} \mid \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I})}{p(\boldsymbol{z}_{1:I}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i} - \{t_{i,j,k}\} \mid \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I})} \\ \propto \left(\frac{\Gamma\left(\alpha_k + t_{i,1,k}\right)}{\Gamma\left(\sum_k \alpha_k + \sum_k t_{i,1,k}\right)} \right)^{\delta_{j,1}} \left(\frac{S^{n_{i,j-1,k}+t_{i,j,k}}_{t_{i,j-1,k},a_i}}{(b_i)_{N_{i,j-1}+T'_{i,j}}} \right)^{1-\delta_{j,1}} (b_i|a_i)_{T'_{i,j}} S^{n_{i,j,k}+t_{i,j+1,k}}_{t_{i,j,k},a_i} \tag{7}$$

Now, we can easily estimate the topic distribution $\boldsymbol{\mu}$ and topic–word distribution $\boldsymbol{\Phi}$, from the statistics obtained after the convergence of Markov chain. They can be approximated from the following mean posterior expected values (using the mean of a Dirichlet distribution and the mean of the PDP) via sampling. For the document topic distribution $\boldsymbol{\mu}_{i,0}$, we have

$$\widehat{\mu}_{i,0,k} = \mathbf{E}_{z_i, \boldsymbol{t}_{i,1:J_i} \mid \boldsymbol{w}_i, \boldsymbol{\alpha}, \boldsymbol{\Phi}, a_i, b_i} \left[\frac{\alpha_k + t_{i,0,k}}{\sum_k \alpha_k + \sum_k t_{i,0,k}} \right] \tag{8}$$

And the segment topic distribution $\boldsymbol{\mu}_{i,j}$ $(1 \le j \le J_i)$ can be estimated as

$$\widehat{\mu}_{i,j,k} \\ = \mathbf{E}_{z_i, \boldsymbol{t}_{i,1:J_i} \mid \boldsymbol{w}_i, \boldsymbol{\alpha}, \boldsymbol{\Phi}, a_i, b_i} \left[\frac{a_i T_{i,j} + b_i}{b_i + N_{i,j} + T_{i,j+1}} \mu_{i,j-1,k} + \frac{(n_{i,j,k} + t_{i,j+1,k}) - a_i t_{i,j,k}}{b_i + N_{i,j} + T_{i,j+1}} \right] \tag{9}$$

Then, the topic–word distribution is given by

$$\widehat{\phi}_{k,w} = \mathbf{E}_{z_i, \boldsymbol{t}_{i,1:J_i} \mid \boldsymbol{w}_i, \boldsymbol{\alpha}, \boldsymbol{\Phi}, a_i, b_i} \left[\frac{\gamma_w + M_{k,w}}{\sum_w (\gamma_w + M_{k,w})} \right] \tag{10}$$

The collapsed Gibbs sampling algorithm for our proposed model is outlined in Fig. 4. 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 initialised to 1. Each Gibbs sampler then constitutes applying Eqs. (4), (5) or (6) to every word in the

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

Topic assignment initialisation: randomly initialise the topic assignment for all words.
Table count initialisation: randomly initialise 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 2
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
        Recursively check all table counts, $t_{i,j',k'}$, where $1 \leq j' \leq j$, to make sure $0 \leq t_{i,j',k'} \leq n_{i,j',k'} + t_{i,j'+1,k'}$ holds;
        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. (4), Eq. (5) or Eq. (6) depending on the value of $u_{i,k}$
        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 the new topic $k$ using Eq. (7)
        Update the statistics with the new table count
      end for
    end for
    Update $\boldsymbol{\alpha}$ by Newton-Raphson method
    Sample $b_i$ with adaptive rejection sampling
  end for
end for
```

Fig. 4 Collapsed Gibbs sampling algorithm for the SeqLDA model

document collection; and applying Eq. (7) 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 ($\boldsymbol{t}$)). Now, we pick a number of Gibbs samples at regularly spaced intervals. In this article, we average these samples to obtain the final sample, as done in [29]. This collapsed Gibbs sampling algorithm is easy to implement and requires little memory.

4.3 Estimating hyperparameters

Since the PDP is extremely sensitive to the strength parameters (i.e. $\boldsymbol{b}_{1:I}$), which was observed in our initial experiments, we thus propose an algorithm to sample b_i for each documents using the Beta/Gamma auxiliary variable trick, as those in [8,32,34]. The sampling routine is based on the joint distribution Eq. (3).

We first consider the case when the discount parameter $a = 0$. The posterior for b_i is proportional to

$$\prod_j \frac{b_i^{T_{i,j}} \Gamma(b_i)}{\Gamma(b_i + N_{i,j} + T_{i,j+1})}$$

Introduce an auxiliary variable $q_{i,j} \sim Beta(b_i, N_{i,j} + T_{i,j+1})$ for each segment i, j. Then, the joint posterior distribution for $q_{i,j}$ and b is proportional to

$$a_i^{\sum_j^{J_i} T_{i,j}} \prod_j^{J_i} q_{i,j}^{b-1} (1 - q_{i,j})^{N_{i,j} + T_{i,j+1} - 1} \tag{11}$$

Given sampled values of all the auxiliary variables, we can now sample b_i according to their conditional distributions,

$$\begin{aligned} q_{i,j} &\sim Beta(b_i, N_{i,j} + T_{i,j+1}) \\ b_i &\sim Gamma\left(\sum_j T_{i,j} + 1, \sum_j log(1/q_{i,j})\right) \end{aligned}$$

For the case when $a > 0$, the sampling scheme become a bit more elaborate. Now, the posterior for b_i is proportional to

$$a_i^{\sum_j^{J_i} T_{i,j}} \prod_j \frac{\Gamma(b_i/a_i + T_{i,j})}{\Gamma(b_i/a_i)} \frac{\Gamma(b_i)}{\Gamma(b_i + N_{i,j} + T_{i,j+1})}$$

Introducing the same auxiliary variables, as those for $a = 0$, yields a joint posterior distribution proportional to

$$a_i^{\sum_j^{J_i} T_{i,j}} \prod_j^{J_i} \frac{\Gamma(b_i/a_i) + T_{i,j}}{\Gamma(b_i/a_i)} q_{i,j}^{b_i - 1} (1 - q_{i,j})^{N_{i,j} + T_{i,j+1} - 1} \tag{12}$$

It is easy to show that the above distribution is log concave in b, so we here adopted an adaptive rejection sampling algorithm [13]. Sampling the strength parameter b allows a different value for each document, even for each segment with only a slight modification of Eq. (11) and Eq. (12). In addition, although we did not study the discount parameter a_i in this work, it could also be optimised or sampled.

Instead of using symmetrical Dirichlet prior $\boldsymbol{\alpha}$, we can use a non-symmetrical Dirichlet prior whose components have to be estimated. The estimation algorithms proposed in the literature are base on either maximum likelihood or maximum a posteriori, such as the Moment–Match and the Newton–Raphson iteration. Here, we adopt the Newton–Raphson method following the early work by Minka [22]. According to Eq. (3), the gradient of the log-likelihood is

$$\begin{aligned} \frac{\mathrm{d}f(\boldsymbol{\alpha})}{d\alpha_k} = &\sum_i \left(\Psi\left(\sum_k \alpha_k\right) - \Psi\left(\sum_k \alpha_k + \sum_k t_{i,1,k}\right)\right) \\ &+ \sum_i \left(\Psi\left(\alpha_k + t_{i,1,k}\right) - \Psi\left(\alpha_k\right)\right) \end{aligned}$$

where $\Psi()$ is known as the digamma function that is the first derivative of log gamma function, and $f(\boldsymbol{\alpha})$ is the model log-likelihood parameterised with α, $f(\boldsymbol{\alpha}) \propto log\left(\prod_i \frac{\mathrm{Beta}_K(\boldsymbol{\alpha} + \boldsymbol{t}_{i,1})}{\mathrm{Beta}_K(\boldsymbol{\alpha})}\right)$.

Then, the Hessian of the log-likelihood is

$$\frac{\mathrm{d}f(\boldsymbol{\alpha})}{d\alpha_k^2} = \sum_i \left(\Psi'\left(\sum_k \alpha_k \right) - \Psi'\left(\sum_k \alpha_k + \sum_k t_{i,1,k} \right) \right) + \sum_i \left(\Psi'\left(\alpha_k + t_{i,1,k} \right) - \Psi'\left(\alpha_k \right) \right)$$

$$\frac{\mathrm{d}f(\boldsymbol{\alpha})}{d\alpha_k \, d\alpha_{k'}} = \sum_i \left(\Psi'\left(\sum_k \alpha_k \right) - \Psi'\left(\sum_k \alpha_k + \sum_k t_{i,1,k} \right) \right) \quad \text{where } k \neq k',$$

and $\Psi'()$ is the trigamma function, i.e. the second derivative of gamma function. Now, a Newton iteration can be computed to optimise Dirichlet prior $\boldsymbol{\alpha}$. In our experiments, we interchangeably upgrade $\boldsymbol{b}$ and $\boldsymbol{\alpha}$ after each main Gibbs sampling iteration. For example, we optimise $\boldsymbol{\alpha}$ for the first 300 iterations with $\boldsymbol{b}$ fixed; then, optimise $\boldsymbol{b}$ for the next 300 iterations with $\boldsymbol{\alpha}$ fixed, and so on. As can be seen, we indeed adopt a greedy approach to optimise the two hyperparamters at the same time, which may not give a global optimum.

5 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 (2.4 GHz), though our code is not multi-threaded. Our previous comprehensive experimental results [11] 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, neither does STM. 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. The held-out perplexity measure [29] is employed to evaluate the generalisation capability to the unseen data. 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 typically demonstrate the superiority of SeqLDA in topic evolvement analysis.

5.1 Data sets

The patent dataset (i.e. Pat-1000) has 1,000 patents that are randomly selected from 8,000 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. most frequent 50 words), 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 'Moby Dick' by Herman Melville, also known as 'The Whale'.

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

Table 3 Dataset statistics

	The Prince	Moby Dick	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	

They are split into chapters which are treated as segments, and only stop-words are removed. Table 3 shows the statistics of these datasets.

5.2 Document modelling

We follow the standard way in document modelling to evaluate the per-word predicative perplexity of the SeqLDA model and the LDA model on the Pat-1000 dataset with 20% held out for testing. The perplexity of a collection of documents is formally defined as:

$$\text{perplexity}(\mathcal{D}_{\text{test}}) = \exp\left\{-\frac{\sum_{i=1}^{I}\ln p(\boldsymbol{w}_i)}{\sum_{i=1}^{I}N_i}\right\}$$

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 generalisation capability. In our experiments, it is computed based on the held-out method introduced in [29]. 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 with Eqs. (8)–(10) 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.

We first investigate the performance of our SeqLDA model with or without the hyperparameter estimation proposed in Sect. 4.3. Four sets of experiments[2] have been done. They are, respectively, the SeqLDA model with $\alpha = 0.10$ (i.e. symmetrical $\boldsymbol{\alpha}$), $b = 10$ and $a = 0.2$ (SeqLDA); with $\boldsymbol{\alpha}$ optimised by Newton–Raphson method, $b = 10$ and $a = 0.2$ (SeqLDA_alpha); with $\alpha = 0.10$, b optimised by sampling method and $a = 0.2$ (SeqLDA_b); and with both $\boldsymbol{\alpha}$ and b optimised and $a = 0.2$ (SeqLDA_alpha_b). Note that for simplicity, b is optimised for each document, even though we can optimise b for each segment. Figure 5 shows the results in terms of perplexity.

According to the p-values of the paired t-test (as shown in Table 4), there is no significant difference between the manually optimised SeqLDA model and the automatically optimised models. We have observed that the average value of the optimised asymmetrical α is close to 0.10. The perplexity of the SeqLDA with only alpha optimised becomes lower than others when k is getting larger ($k > 50$). In contrast, the SeqLDA with both α and b optimised yields slightly higher perplexity. This might be because the way that we used to carry out the optimisation is kind of greedy, which cannot reach a global optimum for both α and b.

[2] We have first done a series of experiments with the value of α ranging from 0.01 to 0.90 to manually choose the optimal one, which is 0.10. And, the values of b and a are chosen empirically based on our initial experiments. They are $b = 10$ and $a = 0.20$

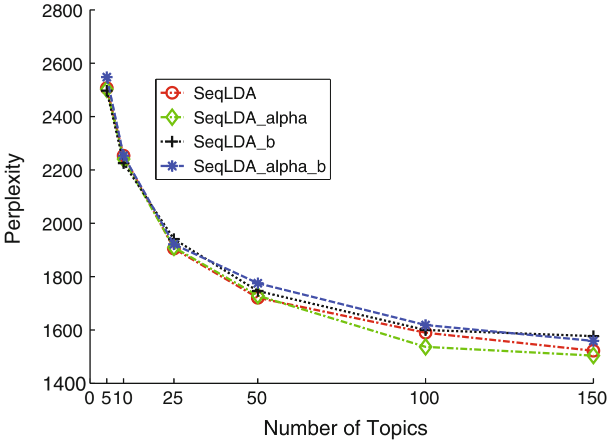


Fig. 5 Perplexity comparison amongst different optimisation approaches on the Pat-1000 dataset with 20% hold out for testing

Table 4 p-values for paired t-test for results in Fig. 5

	Pat-1000		
	SeqLDA_alpha	SeqLDA_b	SeqLDA_alpha_b
SeqLDA	2.2e-1	2.8e-1	1.3e-2

We can therefore conclude that our hyperparameter optimisation algorithms work as well as the manual optimisation. And, we can further claim that these hyperparamters are not difficult to set up in order to get nice results.

In addition, we ran another set of experiments to verify whether there indeed exists a sequential topical dependency amongst segments of each document. Instead of retaining the original layout of segments (i.e. the original order of paragraphs in a patent), we have randomly permuted the order of the segments for both the training dataset and the testing dataset. In Fig. 6, ‘NP’ indicates the seqLDA model trained and tested without permutation, ‘PTrTe’ indicates the model trained and tested with permutation, and ‘PTe’ indicates the model tested with permutation but trained without permutation. Take $k = 25$ as an example, the perplexity corresponding to the original layout (1,905.2) is much lower than that corresponding to the randomly permuted order (2,009.8). Thus, the significant difference shows that the sequential topical structure does exist in the patents, and considering this structure can improve the accuracy of text analysis in terms of the perplexity.

Next, we compare the SeqLDA model with the LDA model (the bench mark model in our view). In order to make a fair comparison, we set hyper-parameters fairly, since they are important for the two models. We employ the moment–match algorithm [22] to optimise $\boldsymbol{\alpha}$ for the LDA model, and fix all parameters for our SeqLDA model as: $a = 0.2$, $b = 10$, $\alpha = 0.1$. And γ is set to $200/W$ for both models. Note that we seek to automatically optimise the parameter settings for the LDA model, which enables us to draw fair conclusions on SeqLDA’s performance.

Figure 7 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

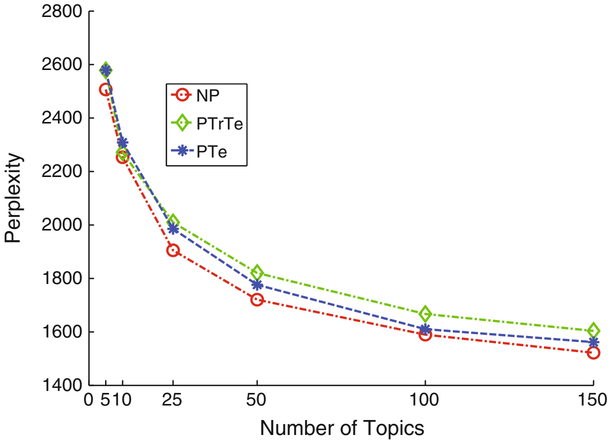


Fig. 6 Perplexity comparison to verify the existence of the sequential topical dependency

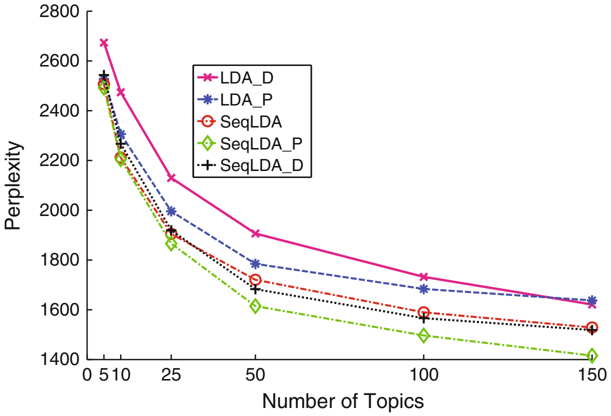


Fig. 7 Perplexity comparison amongst the SeqLDA model and the LDA model on the Pat-1000 dataset

(SeqLDA_D) or LDA_P (SeqLDA_P). The boosting is done by using the topic assignments learnt by the LDA model to initialise 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 5 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.

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 Fig. 8, 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% patents for training and 2,000

Table 5 p-values for paired t-test for results in Fig. 7

	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

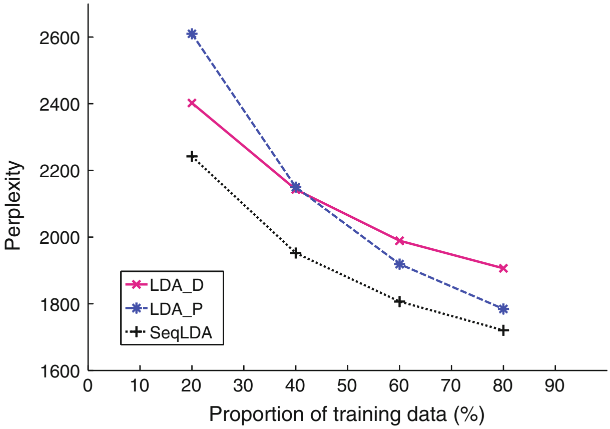


Fig. 8 Perplexity comparison on the Pat-1000 dataset with different percentages of training data ($K = 50$)

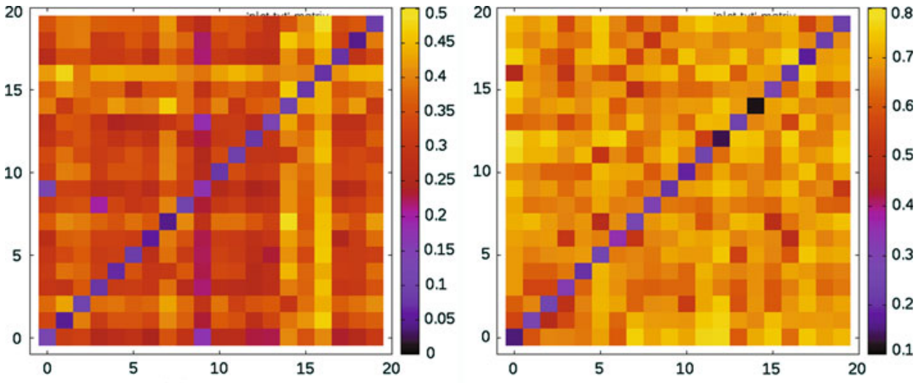
Gibbs iterations, is approximately 5 h for LDA, and 25 h for SeqLDA, which indicates that the SeqLDA is still reasonably manageable in terms of training time.

5.3 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 organises, for instance, her stories in a book or her 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 "Moby Dick".

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. Figure 9a and b show the confusion matrices between the topic distributions generated by the SeqLDA model and the LDA model 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 'Moby Dick'. For 'The Prince', the major confusion is with topic-0 and 9 yielding some blueish off-diagonal.

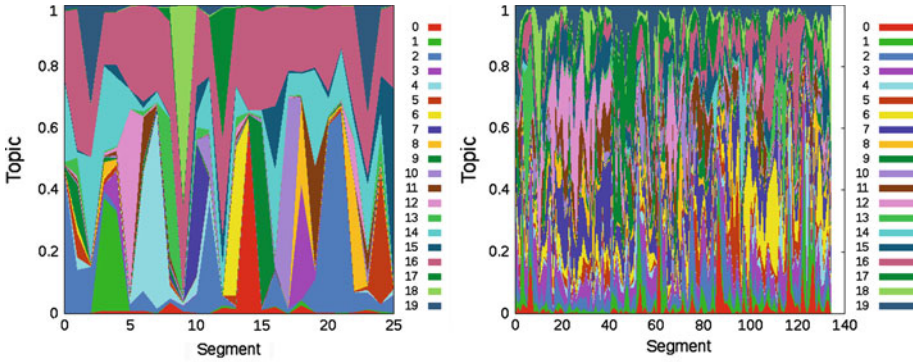
After aligning the topics, we plot the topic distributions (i.e. sub-ideas) as a function of chapter to show how each topic evolves, as shown in Figs. 10 and 11 respectively. Immediately, we see that the topic evolving patterns over chapters learnt by SeqLDA are much clearer that those learnt by LDA. For example, compare the subfigures in these two figures, it is hard to find the topic evolvement patterns in Fig. 10b learnt by LDA; in contrast, we can find the



(a) The Prince

(b) Moby Dick

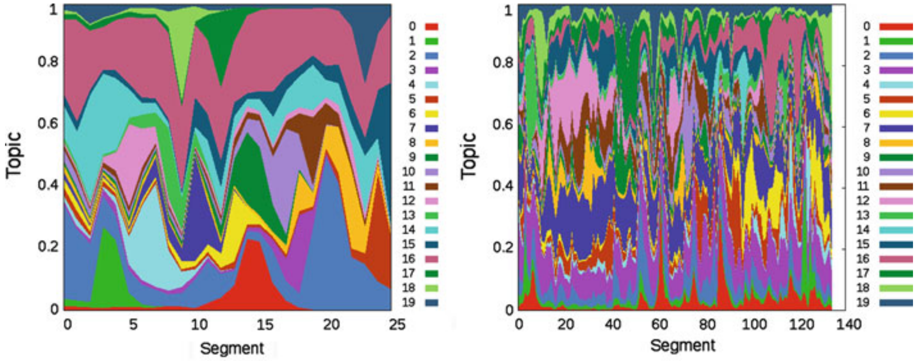
Fig. 9 Topic alignment by confusion matrix



(a) The Prince

(b) Moby Dick

Fig. 10 Topic evolvement analysis by LDA

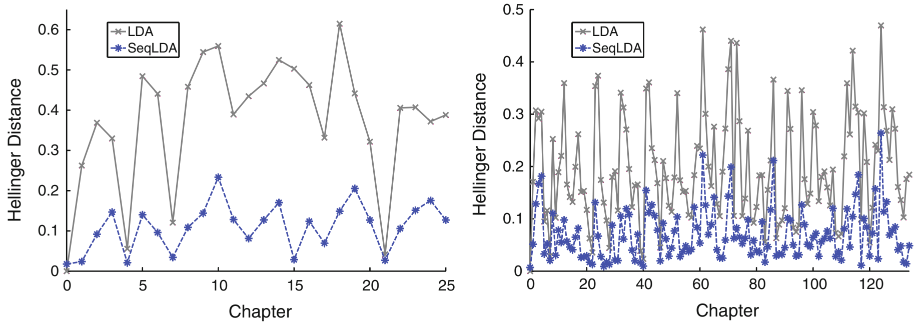


(a) The Prince

(b) Moby Dick

Fig. 11 Topic evolvement analysis by SeqLDA

patterns in Fig. 11b, 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 Chaps. 15–40 and again around Chaps. 65–70, which is coherent with the book.



(a) The Prince **(b)** Moby Dick

Fig. 12 Topic evolvement by Hellinger Distance

Table 6 Typical topics learnt from 'The Prince'

LDA	Topic-0	Servant servants pandolfo good opinion cares honours recognise honest comprehends venafro trust attention fails praise judgment honouring form thinking correct error clever choosing rank disposed prime useless Since a faithfull study
	Topic-9	Truth emperor flatterers opinions counsel's wisdom contempt advice listen preserved bold counsel resolutions speaking maximilian 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 maximilian choosing advantageous listen thinking capable recognise
	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

Top 30 words are listed as examples

Moreover, Fig. 12a and 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 Chaps. 7–10 in Fig. 12a. 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 readers understand more precisely how a book is organised.

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 Fig. 9a), also show in Table 6. 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 Chap. 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 titles. Now consider the issue of topic-0 and 9. Inspection shows topic-9 learnt by LDA occurring in Chaps. 2 and 16 is split into two by SeqLDA: the Chap. 16 part joins topic-0 which has its strength in the neighbouring Chap. 15, and the topic-0 part broadens out amongst the three Chaps. 1–3. These topics are illustrated in Table 6 and it can be seen that topic-0 and topic-9 by LDA talk about related themes.

Now consider 'Moby Dick' 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 Chap. 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 Fig. 11b.

6 Conclusion

In this article, we have proposed a novel generative model, the sequential latent Dirichlet allocation (SeqLDA) model by explicitly considering the document structure in the hierarchical modelling. The sequential topical dependencies buried in the higher level of document structure are captured by the dependencies amongst the segments' subjects (or ideas) which are further approximated by topic distributions. Thus, the topic evolvements can be estimated by observing how topic distributions change amongst segments. Unlike other Markov chain-based models, the SeqLDA model detects the rise and fall of topics within each individual document by putting the Markov assumption on the topic distributions.

We have developed for the SeqLDA model an efficient collapsed Gibbs sampling algorithm based on the hierarchical two-parameter Poisson–Dirichlet process (HPDP) on top of the corresponding Chinese Restaurant Process. Instead of sampling the full customer seating arrangement, our algorithm introduces an auxiliary latent variable, i.e. table count, to sum out the exact customer partitions in the restaurants. In this way, the real valued parameter of the PDP can easily be integrated out. Having observed the PDP is sensitive to the strength parameters (i.e. b), we proposed an adaptive rejection sampling method to optimise b. 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 amongst segments within a document) has been demonstrated in our experiments. The experimental results also indicate that the document structure can aid in the statistical text analysis, and structure-aware topic modelling approaches provide a solution 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, text segmentation or document classifications, where sequential structures could play an important role. The two parameters a and b in the PDP can be optimised dynamically for each segment, instead of fixed for each corpus or each document in order to handle sizeable topic drift between segments i.e. where the correlations between two successive segments are not very strong. The SeqLDA model currently uses the first-order Markov chain to tie a sequence of LDA models, which means the topic distribution of a segment is drawn from the PDP with that of its preceding segment as a basis. The Markov chain could be extended to a more general graph, for example, by taking both document topic distribution and preceding segment topic distribution as a base distribution in the PDP.

Acknowledgments We thanks anonymous reviewers for their very useful comments and suggestions. 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 programme. Dr. Huidong Jin was partly supported by CSIRO's Water for a Healthy Country Flagship for this work.

Appendix A: Two-parameter Poisson–Dirichlet process and Chinese restaurants

The two-parameter Poisson–Dirichlet process (PDP), is a generalisation 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 $\boldsymbol{\mu}_{j-1}$ [32].

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 $\boldsymbol{\mu}$ marginalised out is specified as follows: the first customer sits at the first table; the $(n+1)$th subsequent customer sits at the tth 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 tth 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 $k \in \{1, \ldots, K\}$. This finite discrete case has some attractive properties shown in [9], which follows some earlier work of [32]. 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 $j = 1, \ldots, J$

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

where $M_j = \sum_k m_{j,k}$. Introduce auxiliary 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

$$p(\boldsymbol{n}_j, \boldsymbol{t}_j | a, b, \boldsymbol{\mu}_{j-1}) = C_{\boldsymbol{n}_j}^{M_j} \frac{(b|a)_{\sum_k t_{j,k}}}{(b)_{M_j}} \prod_k S_{t_{j,k},a}^{m_{j,k}} \prod_k \mu_{j-1,k}^{t_{j,k}} \tag{13}$$

where $C_{\boldsymbol{n}_j}^{M_j}$ is the multi-dimensional choose function of a multinomial.

Note that in the hierarchical 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 Eq. (13). Thus, there are two types of customers in each restaurant using the notation of Table 2, the ones arriving by themselves ($\boldsymbol{n}_j$), and those sent by its child restaurant ($\boldsymbol{t}_{j+1,k}$).

Appendix B: The derivation of the joint distribution

To show the derivation of Eqs. (2) and (3), we begin by calculating $p(\boldsymbol{w}|\boldsymbol{z})$ with the use of $p(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\Phi})$. It can be derived from a multinomial on the observed word counts given the associated topics as following

$$p(\boldsymbol{w}_{1:I} \mid \boldsymbol{z}_{1:I}, \boldsymbol{\Phi}) = \prod_i \prod_k \prod_w \phi_{k,w}^{M_{i,k,w}} = \prod_k \prod_w \phi_{k,w}^{M_{k,w}}$$

which can give the last product in Eq. (2). The target distribution $p(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\gamma})$ is obtained by integrating over $\boldsymbol{\Phi}$, which can be done componentwise using Dirichlet integrals within the product over topic z:

$$\begin{aligned}
p(\boldsymbol{w}_{1:I} \mid \boldsymbol{z}_{1:I}, \boldsymbol{\gamma}) &= \int p(\boldsymbol{w}_{1:I} \mid \boldsymbol{z}_{1:I}, \boldsymbol{\Phi}) p(\boldsymbol{\Phi} \mid \boldsymbol{\gamma}) d\boldsymbol{\Phi} \\
&= \int \prod_k \prod_w \phi_{k,w}^{M_{k,w}} \frac{1}{\text{Beta}_W(\boldsymbol{\gamma})} \prod_w \phi_{k,w}^{\gamma_w - 1} \, d\boldsymbol{\phi}_k \\
&= \int \prod_k \underbrace{\frac{1}{\text{Beta}_W(\boldsymbol{\gamma})} \prod_w \phi_{k,w}^{M_{k,w}+\gamma_w-1}}_{\text{Dirichlet formulation}} \, d\boldsymbol{\phi}_k \\
&= \prod_k \frac{\text{Beta}_W(\boldsymbol{\gamma} + \boldsymbol{M}_k)}{\text{Beta}_W(\boldsymbol{\gamma})}
\end{aligned} \tag{14}$$

Then, the challenge here is to calculate the topic distribution $p(\boldsymbol{z}|\boldsymbol{\alpha})$ with the involvement of networks of the PDP, refer to 'Appendix A'. Instead of drawing topic z for the corresponding segment topic distribution $\boldsymbol{\mu}_{i,j}$, where $j > 0$, we sample z directly from the base distribution, i.e. document topic distribution $\boldsymbol{\mu}_{i,0}$, by marginalising out $\boldsymbol{\mu}_{i,j}$. Since the PDP is self-conjugate in the finite discrete space, the marginalization can be done in a recursive fashion in the CRP configuration, as discussed in 'Appendix A' and shown in Fig. 3. Using Dirichlet integrals and Eq. (13), we have (with the new auxiliary variable, table counts $\boldsymbol{t}$)

$$\begin{aligned}
&p(\boldsymbol{z}_{1:I}, \boldsymbol{t}_{1:I} \mid \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I}) \\
&= \int\int \prod_i p(\boldsymbol{\mu}_{i,0} \mid \boldsymbol{\alpha}) \prod_{j=1}^{J_i} \underbrace{p(\boldsymbol{\mu}_{i,j} \mid a_i, b_i, \boldsymbol{\mu}_{i,j-1})}_{\boldsymbol{\mu}_{i,j} \sim \text{PDP}(a_i, b_i, \boldsymbol{\mu}_{i,j-1})} \prod_{l=1}^{L_j} p(z_{i,j,l} \mid \boldsymbol{\mu}_{i,j}) \, d\boldsymbol{\mu}_{i,j} d\boldsymbol{\mu}_{i,0}
\end{aligned}$$

$$= \int \prod_i \frac{1}{\mathrm{Beta}_K(\boldsymbol{\alpha})} \prod_k \mu_{i,0,k}^{\alpha_k - 1} \underbrace{\left(\prod_j \frac{(b_i|a_i)_{T_{i,j}}}{(b_i)_{N_{i,j}+T_{i,j+1}}} \prod_{j,k} S^{n_{i,j,k}+t_{i,j+1,k}}_{t_{i,j,k},a_i} \right) \prod_k \mu_{i,0,k}^{t_{i,1,k}}}_{\text{marginalise out } \mu_{i,j} \text{ recursively, for } j>0} d\boldsymbol{\mu}_{i,0}$$

$$= \int \prod_i \underbrace{\frac{1}{\mathrm{Beta}_K(\boldsymbol{\alpha})} \prod_k \mu_{i,0,k}^{\alpha_k + t_{i,1,k} - 1}}_{\text{Dirichlet formulation}} \left(\prod_j \frac{(b_i|a_i)_{T_{i,j}}}{(b_i)_{N_{i,j}+T_{i,j+1}}} \prod_{j,k} S^{n_{i,j,k}+t_{i,j+1,k}}_{t_{i,j,k},a_i} \right) d\boldsymbol{\mu}_{i,0}$$

$$= \prod_i \frac{\mathrm{Beta}_K\left(\boldsymbol{\alpha} + \boldsymbol{t}_{i,1}\right)}{\mathrm{Beta}_K\left(\boldsymbol{\alpha}\right)} \prod_{i,j} \frac{(b_i|a_i)_{T_{i,j}}}{(b_i)_{N_{i,j}+T_{i,j+1}}} \prod_{i,j,k} S^{n_{i,j,k}+t_{i,j+1,k}}_{t_{i,j,k},a_i} \tag{15}$$

The joint distribution therefore becomes the combination of Eqs. (14) and (15), which is Eq. (3).

References

1. Ahmed A, Xing EP (2010) Timeline: a dynamic hierarchical Dirichlet process model for recovering birth/death and evolution of topics in text stream. In: Proceedings of the twenty-sixth conference annual conference on uncertainty in artificial intelligence'
2. AlSumait L, Barbará D, Domeniconi C (2008) On-line LDA: adaptive topic models for mining text streams with applications to topic detection and tracking. In: Proceedings of the eighth international conference on data mining, pp 3–12
3. Blei D, Lafferty J (2006a) Correlated topic models. In: Advances in neural information processing systems, vol 18, pp 147–154
4. Blei DM, Griffiths TL, Jordan MI (2010) The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. J ACM 57(2):1–30
5. Blei DM, Lafferty JD (2006b) Dynamic topic models. In: Proceedings of the 23rd international conference on Machine learning, pp 113–120
6. Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. J Mach Learn Res 3:993–1022
7. Blei D, McAuliffe J (2007) Supervised topic models. In: Advances in neural information processing systems, vol 20, pp 121–128
8. Buntine W, Du L, Nurmi P (2010) Bayesian networks on Dirichlet distributed vectors. In: Proceedings of the fifth European workshop on probabilistic graphical models (PGM-2010), pp 33–40
9. Buntine W, Hutter M (2010) A bayesian review of the poisson–dirichlet process, Technical Report arXiv:1007.0296, NICTA and ANU, Australia. http://arxiv.org/abs/1007.0296
10. Buntine W, Jakulin A (2006) Discrete components analysis, In: Subspace, latent structure and feature selection techniques. Springer, Berlin
11. Du L, Buntine W, Jin H (2010) A segmented topic model based on the two-parameter Poisson–Dirichlet process. Mach Learn 81:5–19
12. Du L, Buntine WL, Jin H (2010b) Sequential latent Dirichlet allocation: discover underlying topic structures within a document. In: Proceedings of the 2010 IEEE international conference on data mining. ICDM '10, pp 148–157
13. Gilks WR, Wild P (1992) Adaptive rejection sampling for Gibbs sampling. J R Stat Soc Ser C 41(2): 337–348
14. Griffiths TL, Steyvers M (2004) Finding scientific topics. Proc Natl Acad Sci USA 101(1):5228–5235
15. Griffiths TL, Steyvers M, Blei DM, Tenenbaum JB (2005) Integrating topics and syntax. In: Advances in neural information processing systems, vol 17, pp 537–544
16. He Q, Chen B, Pei J, Qiu B, Mitra P, Giles L (2009) Detecting topic evolution in scientific literature: how can citations help?. In: Proceeding of the 18th ACM conference on information and knowledge management, pp 957–966
17. Hofmann T (1999) Probabilistic latent semantic indexing. In: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 50–57

18. Ishwaran H, James LF (2001) Gibbs sampling methods for stick breaking priors. J Am Stat Assoc 96:161–173
19. Kandylas V, Upham S, Ungar L (2008) Finding cohesive clusters for analyzing knowledge communities. Knowl Inform Syst 17:335–354
20. Li T (2008) Clustering based on matrix approximation: a unifying view. Knowl Inform Syst 17:1–15
21. Mimno D, McCallum A (2008) Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. In: Proceedings of the twenty-fourth conference annual conference on uncertainty in artificial intelligence, pp 411–418
22. Minka TP (2000) Estimating a Dirichlet distribution. Technical report, MIT
23. Nallapati RM, Ditmore S, Lafferty JD, Ung K (2007) Multiscale topic tomography. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 520–529
24. Newman D, Asuncion A, Smyth P, Welling M (2008) Distributed inference for latent Dirichlet allocation. In: Advances in neural information processing systems, vol 20, pp 1081–1088
25. Peng W, Li T (2011) Temporal relation co-clustering on directional social network and author-topic evolution. Knowl Inform Syst 26:467–486
26. Pitman J, Yor M (1997) The two-parameter Poisson–Diriclet distribution derived from a stable subordinator. Ann Prob 25(2):855–900
27. Porteous I., Newman D., Ihler A., Asuncion A., Smyth P, Welling M (2008) Fast collapsed Gibbs sampling for latent Dirichlet allocation. In: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 569–577
28. Ren L, Dunson DB, Carin L (2008) The dynamic hierarchical dirichlet process. In: Proceedings of the 25th international conference on machine learning, pp 824–831
29. Rosen-Zvi M, Griffiths T, Steyvers M, Smyth P (2004) The author-topic model for authors and documents. In: Proceedings of the 20th conference on Uncertainty in Artificial Intelligence. AUAI Press, Arlington, Virginia, United States, pp 487–494
30. Shafiei MM, Milios EE (2006) Latent Dirichlet co-clustering. In: Proceedings of the sixth international conference on data mining, pp 542–551
31. Shen ZY, Sun J, Shen YD (2008) Collective latent Dirichlet allocation. In: Proceedings of the 2008 eighth IEEE international conference on data mining. IEEE Computer Society, Washington, DC, pp 1019–1024
32. Teh Y (2006a) A Bayesian interpretation of interpolated Kneser-Ney, Technical Report TRA2/06, School of Computing. National University of Singapore
33. Teh YW (2006b) A hierarchical Bayesian language model based on Pitman-Yor processes. In: Proceedings of the 21st international conference on computational linguistics and the 44th annual meeting of the association for computational linguistics, pp 985–992
34. Teh Y, Jordan MI, Beal MJ, Blei DM (2006) Hierarchical Dirichlet processes. J Am Stat Assoc 101
35. Thurau C, Kersting K, Wahabzada M, Bauckhage C (2010) Convex non-negative matrix factorization for massive datasets. Knowl Inform Syst. doi:10.1007/s10115-010-0352-6
36. Wang C, Blei D, Heckerman D (2008) Continuous time dynamic topic models. In: Proceedings of the 24th annual conference on uncertainty in artificial intelligence, pp 579–586
37. Wang H, Huang M, Zhu X (2008) A generative probabilistic model for multi-label classification. In: Proceedings of the 2008 eighth IEEE international conference on data mining. IEEE Computer Society, Washington, DC, pp 628–637
38. Wang X, McCallum A (2006) Topics over time: a non-Markov continuous-time model of topical trends. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, pp 424–433
39. Wei X, Sun J, Wang X (2007) Dynamic mixture models for multiple time series. In: Proceedings of the 20th international joint conference on artifical intelligence. Morgan Kaufmann Publishers Inc., pp 2909–2914
40. Zhang J, Song Y, Zhang C, Liu S (2010) Evolutionary hierarchical dirichlet processes for multiple correlated time-varying corpora. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 1079–1088

Author Biographies

Lan Du received a B.Sc. degree from the school of Computer Science, Engineering and Mathematics, the Flinders University of South Australia, in 2006 and a first-class honours degree from the College of Engineering and Computer Science, the Australian National University (ANU), in 2007. He is currently a PhD student at ANU College of Engineering and Computer Science. He is also affiliated as a graduate researcher with the Statistical Machine Learning Group at NICTA Canberra research laboratory. His research interests include hierarchical Bayesian reasoning (either parametric or non-parametric), topic modelling, information retrieval, data mining, etc.

Wray Buntine joined NICTA in Canberra Australia in April 2007. He was previously of University of Helsinki and HIIT from 2002, where he conceived and coordinated the EU STREP project ALVIS, for semantic search engines. He was previously at NASA Ames Research Center, UC Berkeley, and Google, and he is known for his theoretical and applied work on data mining and machine learning for document and text analysis, and probabilistic methods generally. He is a Principle Researcher working on applying probabilistic methods to tasks such as information retrieval and text analysis. In 2009 he was co-chair of ECMLPKDD in Bled, Slovenia and is organising a PASCAL2 Summer School on Machine Learning in Singapore in 2011. He reviews for conferences such as CIKM, ECMLPKDD, ICML, KDD, SIGIR, UAI and WWW and is on the editorial board of Data Mining and Knowledge Discovery and Statistics and Computing.

Huidong Jin received the B.Sc. degree from the Department of Applied Mathematics and the M.Sc. degree from the Institute of Information and System Sciences from Xi'an Jiaotong University, PR China, in 1995 and 1998, respectively, and the PhD degree in Computer Science and Engineering from the Chinese University of Hong Kong, Hong Kong, in 2002. He is a senior research scientist, CSIRO Mathematics, Informatics and Statistics, Australia, and an adjunct lecturer, the Australian National University, Australia. Before that, he was with NICTA Canberra Lab, Australia, and Lingnan University, Hong Kong. His research interests include data mining, statistical modelling and learning, optimisation as well as their applications. He is a member of the IEEE and Statistical Society of Australia, and serves on programme committee of various conferences/workshops.

Changyou Chen received his B.S. and M.S. degree in 2007 and 2010 respectively, both from School of Computer Science, Fudan University, Shanghai, China. Now he is a PhD candidate at the College of Engineering and Computer Science, the Australian National University, under supervised by Dr. Wray Buntine. His current research interests include statistical machine learning, graphical models, stochastic processes, and applying them for topic models and related applications.