A Segmented Topic Model based on the Two-parameter Poisson-Dirichlet Process*

Lan Du[1,2], Wray Buntine[2,1], and Huidong Jin[3,1]

[1] Research School of Information Sciences and Engineering
The Australian National University, Canberra, Australia
[2] NICTA, Canberra, Australia
[3] CSIRO Mathematics, Informatics and Statistics, Canberra, Australia
{Lan.Du,Wray.Buntine}@nicta.com.au
Warren.Jin@csiro.au

Abstract. Documents come naturally with structure: a section contains paragraphs which itself contains sentences; a blog page contains a sequence of comments and links to related blogs. Structure, of course, implies something about shared topics. In this paper we take the simplest form of structure, a document consisting of multiple segments, as the basis for a new form of topic model. To make this computationally feasible, and to allow the form of collapsed Gibbs sampling that has worked well to date with topic models, we use the marginalized posterior of a two-parameter Poisson-Dirichlet process (or Pitman-Yor process) to handle the hierarchical modelling. Experiments using either paragraphs or sentences as segments show the method significantly outperforms standard topic models on either whole document or segment, and previous segmented models, based on the held-out perplexity measure.

1 Introduction

Documents come with structure: a section contains paragraphs which itself contains sentences; a blog page contains a sequence of comments and links to related blogs; a paper contains appendices and references to related work. Some forms of structure are modelled with links in a document, and many different approaches follow from the key initial paper here [1]. Some forms of structure are readily modelled simply by typing tokens, separating out the words, the links, maybe the names, into different multinomials in the topic model, easily done with existing theory [2, Section 5.2]. Other forms of structure work with the topic space themselves [3, 4]. However, a different challenge in text analysis is the problem of understanding the document structure. In this paper, we look at the original layout of each document as our guide to structure by following the ideas of Shafiei and Milios [5], who developed a hierarchical model of the segments in a document.

Given a collection of documents, each of which consists of a set of segments (*e.g.*, sections, paragraphs, or sentences), each segment contains a group of words,

* Appearing in ECML PKDD 2010, Barcelona.

we wish to explore the latent topic structure of each document by taking into account segments and their layout. We believe segments in a document not only have meaningful content but also provide preliminarily structural information, which can aid in the analysis of the original text. This idea actually originates from the way in which people normally compose documents (*e.g.* essays, theses or books). Obviously, to write a document, we need first come up with some main ideas, then organize segments around them, and the ideas for segments could vary around the main ideas.

We take essay writing as an example. An easily accessible and understandable structure is very important for an essay. Generally, an essay should have main ideas which indicate what the essay deals with; then paragraphs, basic structural units in an essay, are organized around the main ideas. Furthermore, each paragraph should have one or more ideas, called sub-ideas in our work, which must link to the main ideas. It means they are not isolated, but sub-ideas can be more specific than the main ideas, and generally be variations of them. The layout and progression of ideas give the meaningful structure of an essay.

Can we statistically model documents in this manner? We adopt probabilistic generative models called topic models. The basic idea is that each document is a random mixture over several latent topics, each of which is a distribution over words. Topic models specify a simple probabilistic process by which words can be generated. Here, we can consider Latent Dirichlet Allocation (LDA) model, proposed by Blei et al. [6], as a way of modelling "ideas" with topics. However, LDA cannot simultaneously learn main ideas and sub-ideas under the same latent topic settings.

Extending LDA to involve segments of a document, Shafiei and Milios [5] presented a Latent Dirichlet Co-Clustering (LDCC) model. It assumes there are two kinds of topics, *document-topics* (*i.e.*, distributions over segments) and *word-topics* (*i.e.*, distributions over words). Thus the LDCC model does not share topics between documents and their segments. It is also assumed that each segment is associated with only one *document-topic*. We will argue that these assumptions can be removed by using distributions over topics (*i.e.*, topic proportions).

There are other topic models that discover the hierarchical structure of topics, for instance using the Hierarchical Dirichlet Process (HDP) [7], Hierarchical LDA (HLDA) [3], and Pachinko Allocation Model (PAM) [4]. HDP is built on data that have been pre-clustered into a hierarchical structure. HLDA organizes topics into a tree based on the nested Chinese restaurant process (CRP), then generates documents by selecting topics along the paths in the tree. PAM uses a directed acyclic graph (DAG) to model the topic hierarchies. These models attempt to capture the intra-topic correlation (*i.e.*, the hierarchical structure of topics) that is quite different from the document structure we deal with. Another model, Dynamic Topic Models (DTM) [8], analyzes the time evolution of topics among document collections, rather than inside each document.

In this paper, we develop a simple structure topic model using the two-parameter Poisson Dirichlet process (PDP) [9, 10], based on recent theoretical

results of the PDP for finite discrete cases [11]. This has the advantage of allowing a collapsed Gibbs sampler to be developed for the hierarchical structure model. The rest of this paper is organized as follows. In Section 2, we present our new Segmented Topic Model (STM), and then elaborate an approximate inference algorithm for STM in Section 3. STM is compared with previous models in Section 4, and experiments based on unbiased evaluations reported in Section 5. Our experiments clearly illustrate the superiority of our STM over previous models.

2 Segmented Topic Model

Our Segmented Topic Model (STM) is a four-level probabilistic generative topic model: two levels of topics proportions, a level of topics and a level of words.

Before specifying STM, we list all notations and terminologies used in this paper. Notations are depicted in Table 1. We define the following terms and dimensions:

- A *word* is the basic unit of our data, indexed by $\{1, \cdots, W\}$.
- A *segment* is a sequence of L words. It can be a section, paragraph, or even sentence. In this work, we assume segments are paragraphs or sentences.
- A *document* is an assemblage of J segments, as shown in Figure 1(b).
- A *corpus* is a collection of I documents.

The basic idea of STM is to assume that each document i has a certain mixture of latent topics, denoted by probability vector $\boldsymbol{\mu}_i$, and is composed of meaningful segments; each of these segments also has a mixture over the same space of latent topics as those for the document, and these are denoted by probability vector $\boldsymbol{\nu}_{i,j}$ for segment j of document i. Both the main ideas of a document and sub-ideas of its segments are modelled here by these distributions over topics. Sub-ideas are taken as variants of the main ideas, and thus sub-ideas can be linked to the main ideas, giving correlations between a document and its segments.

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
$\boldsymbol{\alpha}$	base distribution for document topic probabilities
$\boldsymbol{\mu}_i$	document topic probabilities for document i, base distribution for segment topic probabilities
$\boldsymbol{\nu}_{i,j}$	segment topic probabilities 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

How do the segment proportions $\boldsymbol{\nu}_{i,j}$ vary around the document proportions $\boldsymbol{\mu}_i$? The use of the PDP distribution as $\boldsymbol{\nu}_{i,j} \sim \text{PDP}(a, b, \boldsymbol{\mu}_i)$ distribution is a key innovation here. We would be happy to use, instead, a distribution such as $\boldsymbol{\nu}_{i,j} \sim \mathit{Dirichlet}(b\boldsymbol{\mu}_i)$ where b plays the role of "equivalent sample size". However, such a distribution makes the prior non-conjugate to the likelihood so general MCMC sampling is required and parameter vectors such as $\boldsymbol{\mu}_i$ can no longer be integrated out to yield efficient collapsed Gibbs samplers. We therefore employ the following lemma adapted from [11]:

Lemma 1. *The following approximations on distributions hold*

$$\begin{aligned} &\mathit{PDP}(0, b, \mathit{discrete}(\boldsymbol{\theta})) \approx \mathit{Dirichlet}(b\boldsymbol{\theta}) \ , \\ &\mathit{PDP}(a, 0, \mathit{discrete}(\boldsymbol{\theta})) \approx \mathit{Dirichlet}(a\boldsymbol{\theta}) \qquad (\mathit{as}\ a \to 0), \end{aligned}$$

The first approximation is justified because the means and the first two central moments (orders 2 and 3) of the LHS and RHS distributions are equal. The second approximation is justified because the mean and first two central moments (orders 2 and 3) agree with error $O(a^2)$.

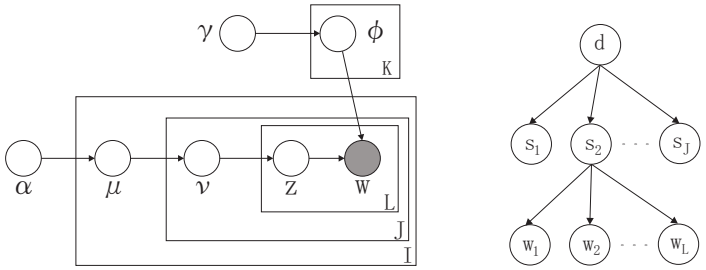
The PDP is a prior conjugate to the multinomial likelihoods, as will be shown in a later section, so allows collapsed Gibbs samplers of the kind used for LDA. Thus, conditioned on the model parameters $\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\Phi}$ and PDP parameters a, b (called *discount* and *strength* respectively), STM assumes the following generative process for each document i:

1. Draw $\boldsymbol{\mu}_i \sim \text{Dirichlet}_K(\boldsymbol{\alpha})$
2. For each segments $j \in \{1, \ldots, J_i\}$
   (a) draw $\boldsymbol{\nu}_{i,j} \sim \text{PDP}(a, b, \boldsymbol{\mu}_i)$
   (b) For each $w_{i,j,l}$, where $l \in \{1, \ldots, L_{i,j}\}$
      i. Select a topic $z_{i,j,l} \sim \text{discrete}_K(\boldsymbol{\nu}_{i,j})$
      ii. Generate a word $w_{i,j,l} \sim \text{discrete}_W(\boldsymbol{\phi}_{z_{i,j,l}})$

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 $\boldsymbol{\Phi}$. The graphical representation of STM is depicted in Figure 1(a), where shaded nodes are observed random variables, unshaded nodes are latent random variables, and the plates indicate repeated sampling. The complete-data likelihood of each document i (*i.e.*, the joint distribution of all observed and latent variables) can be read directly from the graph using the distributions given in the above generative process.

3 Approximate inference by collapsed Gibbs sampling

We have described the motivation behind STM. Here, we elaborate the procedures for inference and parameters estimation under STM. In order to use the model, we need to solve the key inference problem which is to compute the posterior probability of latent variables (*i.e.*, $\boldsymbol{\mu}$, $\boldsymbol{\nu}$ and $\boldsymbol{z}$) given the input $\boldsymbol{\alpha}$, $\boldsymbol{\Phi}$,



(a) STM (b) Doc-structure

Fig. 1. The Segmented Topic Model and the document structure used in this model

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 paragraph j for topic k.
$N_{i,j}$	topic total sum in document i and segment j, *i.e.*, $\sum_k n_{i,j,k}$.
$\boldsymbol{n}_{i,j}$	topic total vector, *i.e.*, $(n_{i,j,1}, ..., n_{i,j,K})$.
$t_{i,j,k}$	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.*, $(t_{i,j,1}, ..., t_{i,j,K})$.

a, b and observations $\boldsymbol{w}$, $p(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{z} \,|\, \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\Phi}, a, b)$. Unfortunately, this posterior distribution cannot be computed directly, due to the intractable computation of marginal probabilities. We must appeal to an approximated inference, where some of the parameters (e.g. $\boldsymbol{\mu}$, $\boldsymbol{\nu}$ and $\boldsymbol{\Phi}$) can be integrated out rather than explicitly estimated. Two standard approximation methods have been applied to topic models: variational inference [6] and collapsed Gibbs sampling [12]. We use the latter taking advantage of the collapsed sampler for the PDP.

Table 2 lists all statistics needed in our algorithm. The statistics $t_{i,j,k}$ and its derivatives are introduced next.

3.1 Marginalizing the PDP

The necessary data for the model is the assignments of words to topics indicated by $z_{i,j,l}$, but also some latent statistics called "table counts of the CRP" indicated by $t_{i,j,k}$, and collectively referred to as *the current state of the CRP*. This subsection explains how these table counts appear. For our purposes, one does not need to know what these table counts are, or how they are derived, since they can be treated as constrained latent variables that just make the sampling

work (according to the lemma below). An explanation of them, however, appears in Appendix A.

The following lemma, which we adapt from [11] is used to handle the PDP. It marginalises the $\boldsymbol{\nu}_{i,j}$ out of the posterior for our model and leaves $\boldsymbol{\mu}_i$ in conjugate form. Consider that part of the complete data likelihood containing segment topic probabilities $\boldsymbol{\nu}_{i,j}$. For each document i, this takes the form $p(\boldsymbol{\nu}_i, \boldsymbol{z}_i|\boldsymbol{\mu}_i, \boldsymbol{\alpha}, \boldsymbol{\Phi}, a, b)$. Ideally, we would like to integrate the vectors $\boldsymbol{\nu}_i$ out for a nice collapsed sampler. We can do this, however, at the cost of introducing the additional latent statistics $\boldsymbol{t}_i$, thus getting $p(\boldsymbol{t}_i, \boldsymbol{z}_i|\boldsymbol{\mu}_i, \boldsymbol{\alpha}, \boldsymbol{\Phi}, a, b)$ where the $\boldsymbol{t}_i$ are integer vectors rather than real-valued vectors. The lemma explains how.

Lemma 2. *Given a probability vector $\boldsymbol{\mu}$ of dimension K, and the following set of priors and likelihoods for $j = 1, ..., J$*

$$\begin{aligned}\boldsymbol{\nu}_j &\sim PDP(a, b, \boldsymbol{\mu})\\ \boldsymbol{n}_j &\sim multinomial_K(\boldsymbol{\nu}_j, N_j)\end{aligned}$$

where $N_j = \sum_k n_{j,k}$, introduce auxiliary latent variables $\boldsymbol{t}_j$ such that $t_{j,k} \leq n_{j,k}$ and $t_{j,k} = 0$ if and only if $n_{j,k} = 0$, then the following marginalised posterior distribution holds

$$p(\boldsymbol{n}_{1:J}, \boldsymbol{t}_{1:J}|a, b, \boldsymbol{\mu}) \;=\; \prod_j C^{N_j}_{\boldsymbol{n}_j} \frac{(b|a)_{\sum_k t_{j,k}}}{(b)_{N_j}} \prod_{j,k} S^{n_{j,k}}_{t_{j,k},a} \prod_k \mu_k^{\sum_j t_{j,k}} \;.$$

The functions introduced in the lemma are as follows: $C^{N_j}_{\boldsymbol{n}_j}$ is the multi-dimensional choose function of a multinomial; $(x)_N$ is given by $(x|1)_N$, $(x|y)_N$ denotes the Pochhammer symbol with increment y, it is defined as

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

where $\Gamma(\cdot)$ denotes the standard gamma function; and $S^N_{M,a}$ is a generalized Stirling number given by the linear recursion [11, 13]

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

for $M \leq N$. It is 0 otherwise and $S^N_{0,a} = \delta_{N,0}$. These rapidly become very large so computation needs to be done in log space using a logarithmic addition.

3.2 The model likelihoods

Consequently, to build a collapsed Gibbs sampler, we first need to derive the marginal distribution over $\boldsymbol{w}$, $\boldsymbol{z}$ and the newly introduced table counts $\boldsymbol{t}$. The Dirichlet priors we put on $\boldsymbol{\mu}_i$ are conjugate to the multinomial distributions, which make the marginalization much easier. Thus, the joint conditional distribution of $\boldsymbol{z}_i, \boldsymbol{t}_{i,1:J_i}, \boldsymbol{w}_i$ can be easily computed by integrating out $\boldsymbol{\mu}_i$, $\boldsymbol{\nu}_{i,j}$ and $\boldsymbol{\Phi}$ respectively as follows:

First, integrating out the segment topic distribution $\boldsymbol{\nu}_{i,j}$ by using Lemma 2, we have $p(\boldsymbol{\mu}_i, \boldsymbol{z}_i, \boldsymbol{w}_i, \boldsymbol{t}_{i,1:J_i} \mid \boldsymbol{\alpha}, \boldsymbol{\Phi}, a, b)$

$$\frac{1}{\mathrm{Beta}_K(\boldsymbol{\alpha})} \prod_k \mu_{i,k}^{\alpha_k + \sum_j t_{i,j,k} - 1} \prod_j \frac{(b|a)_{T_{i,j}}}{(b)_{N_{i,j}}} \prod_{j,k} S_{t_{i,j,k},a}^{n_{i,j,k}} \prod_{w,k} \phi_{k,w}^{M_{i,k,w}}$$

where $\mathrm{Beta}_K(\boldsymbol{\alpha})$ is K dimensional beta function that normalizes the Dirichlet. Then, integrating out the document topic distributions $\boldsymbol{\mu}_i$ and the topic-word matrix $\boldsymbol{\Phi}$, as is usually done for collapsed Gibbs sampling, gives

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

3.3 The collapsed Gibbs sampling algorithm

Collapsed Gibbs sampling is a special form of Markov chain Monte Carlo simulation, which should proceed until the Markov chain has "converged", though in practice we run it for a fixed number of cycles. While the proposed algorithm does not directly estimate $\boldsymbol{\mu}$, $\boldsymbol{\nu}$ and $\boldsymbol{\Phi}$, we will show how they can be approximated using the posterior sample statistics of $\boldsymbol{z}$ and $\boldsymbol{t}$. To apply this algorithm we divides the collapsed Gibbs sampler into two parts. First, given the current table counts $t_{i,j,k}$, we sample the $z_{i,j,l}$ variables. Second, given all assignments of words to topics, we sample the table counts $t_{i,j,k}$ for each topic under each segment.

Now, the full conditional distribution for $z_{i,j,l}$ can be obtained by focusing on a $z_{i,j,l}$, and looking at the proportionalities in Equation (1). For this, $t_{i,j,k}$ is mostly constant, as is $N_{i,j}$. Also, we have to take care with constraints on $t_{i,j,k}$, namely, $t_{i,j,k} \leq n_{i,j,k}$. We should note $t_{i,j,k}$ can be forced to decrease when $n_{i,j,k}$ decreases by removing the current $z_{i,j,l}$. Hereby, to compute the final conditional distribution we have to distinguish between three cases:

1. removing $z_{i,j,l} = k$ forces $n'_{i,j,k} = t'_{i,j,k} = 0$;
2. before removing $z_{i,j,l} = k$, $n_{i,j,k} = t_{i,j,k} > 0$, so we should decrease $t'_{i,j,k} = t_{i,j,k} - 1$;
3. adding $z_{i,j,l}$ forces $n'_{i,j,k} = t'_{i,j,k} = 1$,

where the dash indicates statistics after excluding (or including) the current topic assignment $z_{i,j,l}$. Taking into account all cases, we obtain the final full conditional distribution $p(z_{i,j,l} = k \mid \boldsymbol{z}_{1:I} - \{z_{i,j,l}\}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i}, \boldsymbol{\alpha}, a, b)$

$$\propto \left(\frac{\alpha_k + \sum_j t'_{i,j,k}}{\sum_k \alpha_k + \sum_{j,k} t'_{i,j,k}} \left(b + a T'_{i,j}\right)\right)^{1_{n'_{i,j,k} \equiv 0}} \left(\frac{S_{t'_{i,j,k},a}^{n'_{i,j,k}+1}}{S_{t'_{i,j,k},a}^{n'_{i,j,k}}}\right)^{1_{n'_{i,j,k} > 0}} \frac{\gamma_{w_{i,j,l}} + M'_{k,w_{i,j,l}}}{\sum_w (\gamma_w + M'_{k,w})}$$

Given the current state of topic assignment of each word, the conditional distribution for table count $t_{i,j,k}$ can be obtained by cancelation of terms in Equation (1), yielding $p(t_{i,j,k} \,|\, \boldsymbol{z}_{1:I}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i} - \{t_{i,j,k}\}, \boldsymbol{\alpha}, a, b)$

$$\propto \frac{\Gamma\left(\alpha_k + \sum_j t_{i,j,k}\right)}{\Gamma\left(\sum_k \alpha_k + \sum_{j,k} t_{i,j,k}\right)} (b|a)_{T_{i,j}} S^{n_{i,j,k}}_{t_{i,j,k},a},$$

which stochastically samples the table counts $t_{i,j,k}$ for each restaurant.

From the statistics obtained after the convergence of Markov chain, we can easily estimate the document topic distribution $\boldsymbol{\mu}$, the segment topic distribution $\boldsymbol{\nu}$, and topic-word distributions $\boldsymbol{\Phi}$. They can be approximated from the following posterior expected values via sampling:

$$\widehat{\mu}_{i,k} = \mathbf{E}_{\boldsymbol{z}_i, \boldsymbol{t}_{i,1:J_i} \,|\, \boldsymbol{w}_i, \boldsymbol{\alpha}, \boldsymbol{\Phi}, a, b} \left[\frac{\alpha_k + \sum_j t_{i,j,k}}{\sum_k \alpha_k + \sum_{j,k} t_{i,j,k}} \right] \tag{2}$$

$$\widehat{\nu}_{i,j,k} = \mathbf{E}_{\boldsymbol{z}_i, \boldsymbol{t}_{i,1:J_i} \,|\, \boldsymbol{w}_i, \boldsymbol{\alpha}, \boldsymbol{\Phi}, a, b} \left[\frac{n_{i,j,k} - a \times t_{i,j,k}}{b + N_{i,j}} + \mu_{i,k} \frac{T_{i,j} \times a + b}{b + N_{i,j}} \right] \tag{3}$$

$$\widehat{\phi}_{k,w} = \mathbf{E}_{\boldsymbol{z}_i, \boldsymbol{t}_{i,1:J_i} \,|\, \boldsymbol{w}_i, \boldsymbol{\alpha}, \boldsymbol{\Phi}, a, b} \left[\frac{\gamma_w + M_{k,w}}{\sum_w (\gamma_w + M_{k,w})} \right] \tag{4}$$

3.4 Sampling the Strength Parameter

Initial experiments showed the strength parameter b of the PDP strongly affects perplexity results and seemed difficult to set by optimisation. We therefore developed a simple sampling method using auxiliary variables as follows. Each segment i, j has an auxiliary probability $q_{i,j} \sim \text{Beta}(b, N_{i,j})$. From this, using an improper prior for b of the form $1/b$, the posterior for b is given by

$$b \,|\, \boldsymbol{q}_{1:I}, \boldsymbol{z}_{1:I}, \boldsymbol{w}_{1:I}, \boldsymbol{t}_{1:I,1:J_i}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, a \;\sim\; \text{Gamma}\left(\sum_{i,j} T_{i,j}, \sum_{i,j} \log 1/q_{i,j} \right) . \tag{5}$$

Sampling using these auxiliary variables operates every major Gibbs cycle as follows:

1. Sample $q_{i,j} \sim \text{Beta}(b, N_{i,j})$ for each document i and segment j and compute $\sum_{i,j} \log 1/q_{i,j}$.
2. Sample b according to the condition distribution (5).

4 Comparison with other topic models

In this section we compare STM, in terms of text modelling, with two topic models[4], Latent Dirichlet Allocation (LDA) [6] and Latent Dirichlet Co-Clustering (LDCC) [5].

[4] We have changed some notations from the original papers to make them consistent with ours.

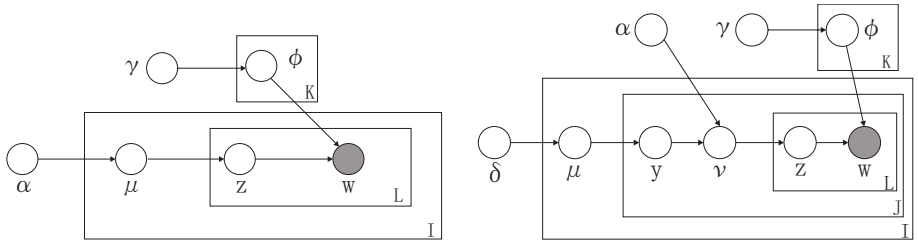
4.1 Latent Dirichlet Allocation

LDA is a three-level probabilistic generative model, the idea of which is that documents are random mixtures over latent topics, where each topic is a distribution over words. It assumes the generative process shown in Figure 2(a), where for each document $\boldsymbol{\mu}_i \sim \text{Dirichlet}_K(\boldsymbol{\alpha})$. Compared with LDA, instead of sampling a topic z directly from the document topic distribution $\boldsymbol{\mu}$, STM adds another layer between z and $\boldsymbol{\mu}$, which is the segment topic distribution $\boldsymbol{\nu}$. Adding this distribution implies a higher fidelity of STM over LDA on modelling the correlation between the document topics and its segment topics (*i.e.*, the topic structure inside a document). LDA could also model the correlation by having two runs through documents and their segments separately. Nevertheless, the consistency of underlying topics between two separate runs cannot be guaranteed, since different runs will come up with different latent topics. Therefore, LDA cannot simultaneously model document topic distributions and segment topic distributions under the same latent topic space, as does our STM.

4.2 Latent Dirichlet Co-clustering

LDCC is a four-level probabilistic model, as STM. It tries to extend LDA by assuming documents are random mixtures over *document-topics*, each of those topics is characterized by a distribution over segments; and segments are random mixtures over *word-topics*, each *word-topic* is a distribution over words. The two different kinds of topics are connected by hyper-parameters $\boldsymbol{\alpha}$, under the assumption that each *document-topic* is a mixture of *word-topics*. It is a kind of nested LDA, as shown in Figure 2(b). LDCC also assumes that each segment is associated with only one *document-topic* (y in Figure 2(b)), which is a quite strong assumption in our view.

In contrast, STM allows documents and segments to share same latent topics, rather than assuming two different kinds, as we believe a document and its segments should be generated from the same kind of topics. Moreover, STM relaxes the assumption on segments by assuming each segment still has a topic distribution drawn from its document topic distribution. Thus, each segment can also exhibit multiple topics, which includes the case that it has only one topic,



(a) LDA (b) LDCC

Fig. 2. The Latent Dirichlet Allocation (LDA) model and the Latent Dirichlet Co-Clustering Model

if the distribution highly concentrates on one topic. In this sense, STM does not have the strong assumptions of LDCC.

5 Experimental Results

We implemented the three models in C, and ran them on a desktop with Intel(R) Core(TM) Quad CPU (2.4GHz), though our code is not multi-threaded. The training time, for instance, on the NIPS dataset with 100 topics and 1000 Gibbs iterations is approximately 5 hours for LDA, 33 hours for LDCC and 20 hours for STM. We first present experimental results for STM, LDA and LDCC on two patent datasets (which will be placed in the UCI Machine Learning Repository). These results present an in depth study of the characteristics of the model. We then present perplexity results on the NIPS dataset[5] and an extract from the Reuters RCV1 corpus [14]. The comparison of the per-word predictive perplexity on held-out testing documents evidently demonstrates the advantage of STM over the other two models.

5.1 Data sets and evaluation criteria

The two patent datasets are randomly selected from 5000 U.S. patents[6] granted between Jan. and Mar. 2009 under the class "computing; calculating; counting" with international patent classification (IPC) code G06. Patents in G06-1000 are split into paragraphs according to the original structure. Patents in G06-990[7] are split into sentences with a Perl package (Lingua::En:Sentence). All stop-words, extremely common words (*e.g.*, top 40 for G06-1000), and less common words (*i.e.*, words appear in less than 5 documents) have been removed. This leads to a vocabulary size of 10385 unique words in G06-1000 and 11518 in G06-990. The G06-1000 dataset contains 1,000 patents, 60,564 paragraphs, and 2,513,087 words. The G06-990 dataset contains 990 patents, 249,102 sentences, and 2,832,364 words. We treat paragraphs or sentences as segments, and hold out 80% of each dataset for training and 20% for testing.

To evaluate the generalization capability of these models to unseen data, we compute perplexity which is a standard measure for estimating the performance of probabilistic language models. The perplexity of a collection $\mathcal{D}$ of I document that is formally defined as: $exp\Big\{-\frac{\sum_{i=1}^{I}\ln p(\boldsymbol{w}_i)}{\sum_{i=1}^{I}N_i}\Big\}$, where $\boldsymbol{w}_i$ indicates all words in document i, and N_i indicates the total number of words in i. A lower perplexity over unseen documents means better generalization capability. In our experiments, it is computed based on the held-out method introduced in [15]. In order to calculate the likelihood of each unseen word in STM, we need to integrate out the sampled distributions (*i.e.* $\boldsymbol{\mu}$, $\boldsymbol{\nu}$, and $\boldsymbol{\Phi}$) and sum over all possible

[5] It is available at `http://nips.djvuzone.org/txt.html`

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

[7] We randomly selected 1000 patents, but 10 were deleted after pre-processing, because they were too small.

topic assignments. Here, we approximate the integrals using a Gibbs sampler and Equations (2), (3) and (4) for each sample of assignments $\boldsymbol{z}, \boldsymbol{t}$.

5.2 Topic variability analysis among segments

We first investigate the variability between topic proportions (*i.e.*, distributions) of documents and those of their segments. As we discussed before, it is modelled by the PDP with two parameters, a and b. Due to space limitations, we only present our studies on how b acting on the diversity among document topic proportions (*i.e.*, $\boldsymbol{\mu}_i$) and their segment topic proportions (*i.e.*, $\boldsymbol{\nu}_{i,j}$). We have observed in our preliminary experiments that b could significantly influence topic proportions. Therefore, we fix $a = 0.2$ for the G06-1000 dataset and $a = 0$ for the G06-990 dataset, change b from 0.1 to 300.0, and then run STM on those two datasets with $k = 50$. The standard deviation (Figure 3(a)) is used to measure the variation of $\boldsymbol{\nu}_{i,j}$, and entropy (Figure 3(c)) to show the expected number of topics in either documents or segments. The prior mean and variance of $\boldsymbol{\nu}_{i,j}$ are given by [11]:

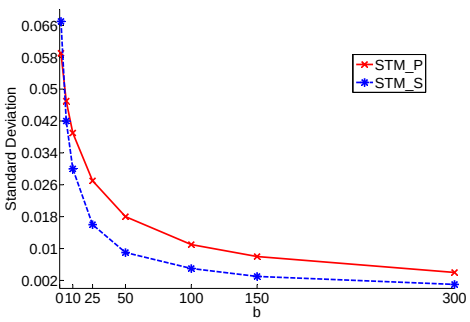
$$\mathcal{E}\left[\boldsymbol{\nu}_{i,j}\right] \;=\; \boldsymbol{\mu}_i \quad ; \qquad \mathcal{V}\left[\boldsymbol{\nu}_{i,j}\right] \;=\; \frac{1-a}{1+b}\left(\text{diagonal}(\boldsymbol{\mu}_i) - \boldsymbol{\mu}_i\boldsymbol{\mu}_i^{\dagger}\right)$$

As shown in Figure 3(a), the standard deviation decreases while b is increasing, as we expect. When b is small, the variance of topic proportions in segments is large. Hereby, the topic proportion $\boldsymbol{\nu}_{i,j}$ of a segment could be quite different from that of the corresponding document ($\boldsymbol{\mu}_i$), as indicated in Figure 3(c) by the different expected number of topics. In contrast, when b gets quite large, the variance of segment topic proportions becomes small. Figure 3(c) shows the expected number of topics in each segment will get close to that of the document it belongs to. In such a case, there could be no difference between a document topic proportion and its segment topic proportions, and segments loose their specificity on topics. We can observe that the perplexity turns out to be larger when b is quite small or quite large in Figure 3(b). Consequently, we can conclude that the topic deviation between a document and its segments should be neither too small nor too big, which somehow complies with the way in which people structure ideas in writing. In addition, Figure 4 lists 6 meaningful topic examples derived from the G06-1000 dataset by our STM trained on 150 topics, with $a = 0.2$ and $b = 10$.

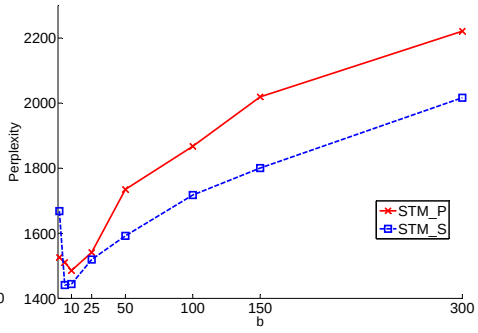
5.3 Perplexity comparison

We follow the standard way in topic modelling to evaluate the per-word predicative perplexity of STM, LDA and LDCC. In the training procedure, each Gibbs sampler is initialized randomly and runs for 500 burn-in iterations. We then draw a total number of 5 samples at a lag of 100 iterations. These samples are averaged to obtain the final trained model, as in [16].

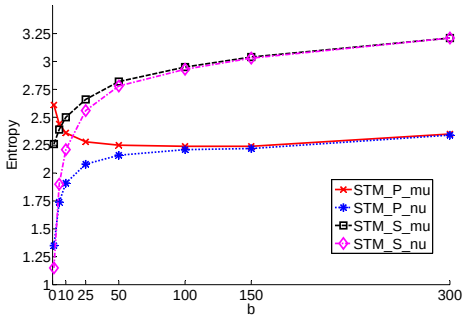
We set hyper-parameters fairly in order to make a scientific comparison, as they are important to these models. Symmetric Dirichlet priors (*i.e.*, $\boldsymbol{\alpha}$ for



(a) Standard deviation w.r.t b



(b) Perplexity w.r.t. b



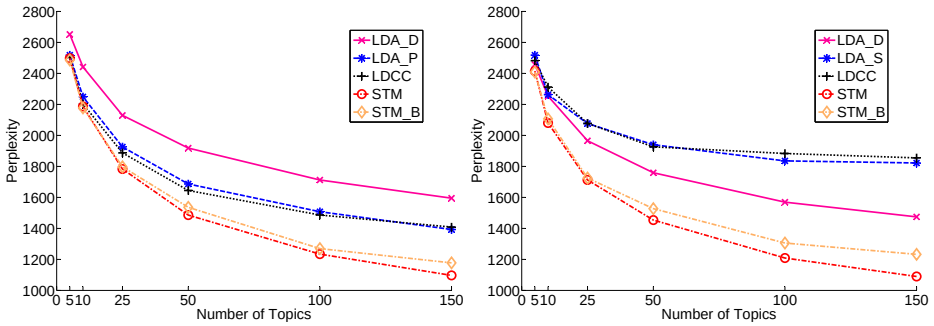
(c) Entropy w.r.t b

Fig. 3. Standard deviation, perplexity and entropy by fixing a and changing b from 0.1 to 300.0. STM_P and STM_S indicate STM runs on paragraphs (G06-1000) and sentences (G06-990) respectively.

T-1	T-2	T-3	T-4	T-5	T-6
security	key	compression	clock	java	word
authentication	keys	compressed	signals	language	string
protected	encryption	encoding	channel	class	words
authorization	encrypted	encoded	timing	objects	character
authorized	content	codes	frequency	environment	frequency
protection	decryption	symbol	channels	library	text
computing	secure	video	synchronization	platform	characters
execution	generated	decoder	generator	native	objects
trusted	secret	decoding	delay	programming	language
permission	public	encoder	enable	applications	prefix

Fig. 4. Topic examples from STM for the G06-1000 dataset

LDA and STM, $\boldsymbol{\delta}$ for LDCC) are simply used in our experiments, although we can estimate them from data using, for instance, the moment-match algorithm proposed in [17]. With γ fixed to $200/W$, we run different settings of $\boldsymbol{\alpha}$ and $\boldsymbol{\delta}$ (from 0.01 to 0.9) for different number of topics (i.e. 5, 10, 25, 50, 100, and 150), and empirically choose the optimal parameters for LDA and LDCC. We have observed, for example, the LDA model trained on $\alpha = 0.1$ is always better on both G06-1000 and G06-990 datasets than on other settings, but the LDCC



(a) Perplexity on the G06-1000 dataset (b) Perplexity on the G06-990 dataset

Fig. 5. Perplexity on the G06-1000 dataset and the G06-990 dataset, for LDA, LDCC, and STM

model varied quite a bit (*e.g.*, $\delta = 0.9$ for 25 *word-topics*, $\delta = 0.01$ for 100 *word-topics*). The number of *document-topics* in LDCC is fixed to 20 for all experiments and $\boldsymbol{\alpha}$ is estimated using the moment-match algorithm, as in [5]. We use $\alpha = 0.5$ in STM for all number of topics without tuning, and set $a = 0.2$ and $b = 10$ for both the G06-1000 dataset and the G06-990 dataset. Note that we seek to optimize the parameter settings for the two competitors (LDA and LDCC), which enables us to draw sound conclusion on STM's performance.

Figure 5(a) presents the results for those models on the G06-1000 dataset. LDA has been run on document level (LDA_D) and paragraph level (LDA_P) separately. It is interesting to see that LDA_P is better than LDA_D. LDCC exhibits better performance than LDA_D, but it is only comparable with LDA_P. The paired t-test, shown in Table 3, gives p-value= 0.05 to the slight improvement, which can be rejected at 0.05 significance level. In contrast, STM (with or without sampling b using the scheme of Section 3.4 , indicated by STM and STM_B respectively) consistently performs better than all the other models. The advantage is especially obvious for large numbers of topics. The superiority of STM over LDA and LDCC is statistically significant according to the paired t-test with p-values shown in the third and fourth columns of Table 3.

Similar comparison on the G06-990 dataset is shown in Figure 5(b). We run LDA (indicated by LDA_S), LDCC and STM on the sentence level. The perplexity of LDCC becomes slightly larger than LDA_S when the number of topics is greater than 50. It is comparable to LDA_S, as LDCC *v.s.* LDA_P in Figure 5(a). Interestingly, the performance of either LDA or LDCC on the

Table 3. P-values for paired t-test

	G06-1000			G06-990		
	LDCC	STM	STM_B	LDCC	STM	STM_B
LDA_D	7.0e-5	1.3e-3	5.4e-4	2.9e-2	4.8e-3	2.2e-3
LDA_P/S	5.0e-2	1.5e-2	8.0e-3	3.9e-1	9.1e-3	6.3e-3
LDCC		3.9e-2	2.8e-2		1.1e-2	7.7e-3

sentence level turns out to be much worse than LDA on the document level. However, the paired t-test results in the last two columns of Table 3 show that our STM is statistically better than both LDA and LDCC. STM can certainly retain its good generalization capability even on sparse text. Evidently, the results illustrated in both Figure 5(a) and Figure 5(b) demonstrate that STM can work remarkably well on both the paragraph level and the sentence level.

5.4 Further experiments

In order to further exhibit the advantage of STM, we also ran it on the NIPS dataset and an extract of the Reuters dataset using $a = 0$ and sampling the strength parameter b according to the scheme of Section 3.4. The NIPS dataset is processed to remove bibliography material (everything after "References") and header material (everything before "Abstract") yielding 1,629 documents, 174,474 sentences (as "segments"), and 1,773,365 words. The Reuters articles are extracted from 20-25/8/1996, and the articles in categories CCAT, ECAT and MCAT are dropped yielding 2,640 articles with a total of 38,182 sentences (as "segments") of average length about 11. Again 80% were used for training and 20% for testing. Perplexity results appear in the table 4.

Table 4. Perplexity on the NIPS dataset and the Reuters dataset

	K	STM	LDCC	LDA_D	LDA_S
NIPS	100	1632	2296	1991	2182
	150	1516	2335	1881	2186
Reuters	100	1893	2154	1824	2687

6 Conclusion

In this paper, we have proposed a segmented topic model (STM), a probabilistic generative model of segments based on the two-parameter Poisson Dirichlet process (PDP). We have developed for STM an efficient collapsed Gibbs sampling algorithm to sample from the posterior PDP. The ability of STM to explore correlated segment topics (*i.e.*, the latent topic structure of a document) has been demonstrated in our experiments by the statistically significant improvement in terms of per-word predictive perplexity compared with the standard topic model (LDA) and previous segmented model (LDCC). The primary benefit of our model is that it allows us to simultaneously model document topic distributions and segment topic distributions under the same latent topic space, without separate runs as LDA or introducing different kinds of topics as in LDCC. Though we have restricted ourselves to paragraphs and sentences, STM readily models other segments, like sections and chapters.

There are many ways that the work described here can be extended. Perhaps the most promising extension to our STM is to consider the full segmented structure of documents, such as essay-paragraph-sentence, blog-comments-sentence, *etc.*, since PDPs can be easily extended to full trees, *e.g.*, HLDA [3], and our collapsed sampling method for PDPs still applies. In applications, our model can be applied to, for example, topic-based multi-document summarization [18].

Acknowledgement

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 was partly supported by CSIRO's Water for a Healthy Country Flagship for this work. We would like to thank David Newman for his valuable comments.

References

1. Cohn, D., Hofmann, T.: The missing link - a probabilistic model of document content and hypertext connectivity. In: NIPS 13. (2001)
2. Buntine, W., Jakulin, A.: Discrete components analysis. In: Subspace, Latent Structure and Feature Selection Techniques. Springer-Verlag (2006)
3. Blei, D., Griffiths, T.L., Jordan, M.I., Tenenbaum, J.B.: Hierarchical topic models and the nested Chinese restaurant process. In: NIPS 16. (2004)
4. Mimno, D., Li, W., McCallum, A.: Mixtures of hierarchical topics with Pachinko allocation. In: Proc. of the 24th ICML. (2007) 633–640
5. Shafiei, M.M., Milios, E.E.: Latent Dirichlet co-clustering. In: Proc. of the 6-th ICDM. (2006) 542–551
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3** (2003) 993–1022
7. Teh, Y., Jordan, M.I., Beal, M.J., Blei, D.M.: Hierarchical Dirichlet processes. Journal of the American Statistical Association **101** (2006)
8. Blei, D.M., Lafferty, J.D.: Dynamic topic models. In: Proc. of the 23rd ICML. (2006) 113–120
9. Pitman, J., Yor, M.: The two-parameter Poisson-Diriclet distribution derived from a stable subordinator. Annals of Probability **25**(2) (1997) 855–900
10. Ishwaran, H., James, L.: Gibbs sampling methods for stick-breaking priors. Journal of ASA **96**(453) (2001) 161–173
11. Buntine, W., Hutter, M.: A Bayesian review of the Poisson-Dirichlet process. Submitted to `arXiv` (2010)
12. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proc Natl Acad Sci U S A **101 Suppl 1** (2004) 5228–5235
13. Teh, Y.: A Bayesian interpretation of interpolated Kneser-Ney. Technical Report TRA2/06, School of Computing, National University of Singapore (2006)
14. Lewis, D., Yand, Y., Rose, T., Li, F.: RCV1: A new benchmark collection for text categorization research. Journal of Machine Learning Research **5** (2004) 361–397
15. Rosen-Zvi, M., Griffiths, T., Steyvers, M., Smyth, P.: The author-topic model for authors and documents. In: Proc. of the 20th UAI. (2004) 487–49
16. Li, W., Blei, D., Mccallum, A.: Nonparametric Bayes Pachinko Allocation. In: Proc. of the 23rd UAI. (2007)
17. Minka, T.P.: Estimating a Dirichlet distribution. Technical report, MIT (2000)
18. Arora, R., Ravindran, B.: Latent Dirichlet allocation and singular value decomposition based multi-document summarization. In: Proc. of ICDM '08. (2008) 713–718
19. Teh, Y.W.: A hierarchical Bayesian language model based on Pitman-Yor processes. In: Proc. of the 21st ICCL. (2006) 985–992

A Two-parameter Poisson Dirichlet process

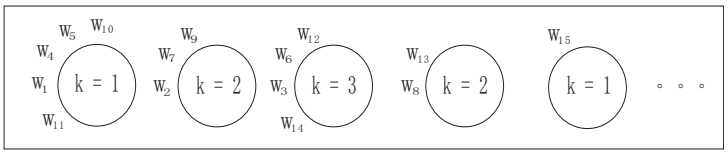


Fig. 6. Analog of Chinese restaurant process for PDP

The two-parameter Poisson-Dirichlet process (PDP), is a generalization of the Dirichlet Process. In regard to STM, let $\boldsymbol{\nu}$ be a distribution over topics (*i.e.* topic proportion) of a segment. We place a PDP prior on $\boldsymbol{\nu}$: $\boldsymbol{\nu} \sim \text{PDP}(a, b, \boldsymbol{\mu})$, where the three parameters are: a base distribution $\boldsymbol{\mu}$ (*i.e.* topic proportion of the document); a ($0 \leq a < 1$) and b ($b > -a$). The strength parameter b can be understood as controlling the amount of variability around $\boldsymbol{\mu}$ [19].

Here, we give a brief discussion of the PDP under the CRP configuration by following the discussion in [11]. Customers in the CRP are words in our model, and dishes in the CRP are topics. Consider a sequence of N customers sitting down in a Chinese restaurant with an infinite number of tables each with infinite capacity. The basic process with $\boldsymbol{\nu}$ 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^*. A snapshot of this process with $n = 15, T = 5, n_1^* = 5, n_2^* = 3, n_3^* = 4, n_4^* = 2, n_5^* = 1$ is shown in Figure 6.

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 [11], which follows some earlier work of [13]. To consider this case we introduce another latent variable: t_k, the *table count* of dish k (referred to as the multiplicity in [11]). In Figure 6 with $n = 15$, for instance, the first and last table have $k = 1$ so the table count for $t_1 = 2$. The table counts are $t_1 = 2, t_2 = 2, t_3 = 1$ and all others zero. Note that $\sum_{k=1}^{K} t_k = T$, and table counts are not observed.