Sampling Table Configurations for the Hierarchical Poisson-Dirichlet Process

Changyou Chen[1,2], Lan Du[1,2], and Wray Buntine[1,2]

[1] Research School of Computer Science,
The Australian National University,
Canberra, ACT, Australia
[2] National ICT, Canberra, ACT, Australia
{Changyou.Chen,Lan.Du,Wray.Buntine}@nicta.com.au

Abstract. Hierarchical modeling and reasoning are fundamental in machine intelligence, and for this the two-parameter Poisson-Dirichlet Process (PDP) plays an important role. The most popular MCMC sampling algorithm for the hierarchical PDP and hierarchical Dirichlet Process is to conduct an incremental sampling based on the Chinese restaurant metaphor, which originates from the Chinese restaurant process (CRP). In this paper, with the same metaphor, we propose a new table representation for the hierarchical PDPs by introducing an auxiliary latent variable, called table indicator, to record which customer takes responsibility for starting a new table. In this way, the new representation allows full exchangeability that is an essential condition for a correct Gibbs sampling algorithm. Based on this representation, we develop a block Gibbs sampling algorithm, which can jointly sample the data item and its table contribution. We test this out on the hierarchical Dirichlet process variant of latent Dirichlet allocation (HDP-LDA) developed by Teh, Jordan, Beal and Blei. Experiment results show that the proposed algorithm outperforms their "posterior sampling by direct assignment" algorithm in both out-of-sample perplexity and convergence speed. The representation can be used with many other hierarchical PDP models.

Keywords: Hierarchical Poisson-Dirichlet Processes, Dirichlet Processes, HDP-LDA, block Gibbs sampler.

1 Introduction

In general machine intelligence domains such as image and text modeling, hierarchical reasoning is fundamental. Bayesian hierarchical modeling of problems is now widely used with applications including n-gram modeling and smoothing [1–3], dependency models for grammar [4, 5], data compression [6], clustering in arbitrary dimensions [7], topic modeling over time [8], and relational modeling [9]. Bayesian hierarchical n-gram models correspond well to versions of Kneser-Ney smoothing [1], the state of the art method in applications and result in competitive string compression algorithms [6]. These hierarchical Bayesian models

D. Gunopulos et al. (Eds.): ECML PKDD 2011, Part I, LNAI 6911, pp. 296–311, 2011.

are intriguing from the probability perspective, as well as sometimes being competitive with performance based approaches. Newer methods and applications are reviewed in [10].

The *two-parameter Poisson-Dirichlet process* (PDP), also referred to as the Pitman-Yor process (named so in [11]), is an extension of the *Dirichlet process* (DP). Related is a particular interpretation of a marginalized version of the model known as the Chinese restaurant process (CRP). The CRP gives an elegant analogy of incremental sampling for these models. These provide the basis of many Bayesian hierarchical modeling techniques. One particular use of the PDP/DP is in the area of topic models where hierarchical PDPs and hierarchical DPs provide elegant machinery for improving the standard simple topic model [12, 13], for instance, with flexible selection of the number of topics using HDP-LDA [14], and allowing document structure to be incorporated into the modeling [15, 16].

This paper proposes a new sampler for the hierarchical PDP based on a new table representation and is organized as follows: Section 2 gives a brief review of the hierarchical Poisson-Dirichlet process. Section 3 then reviews methods for sampling the hierarchical PDP. We present the new table representation for the HPDP in Section 4. A block Gibbs sampler is developed in Section 5, where we also apply our block Gibbs sampler to the HDP-LDA model. Experiment results are reported in Section 6.

2 The Hierarchical Poisson-Dirichlet Process

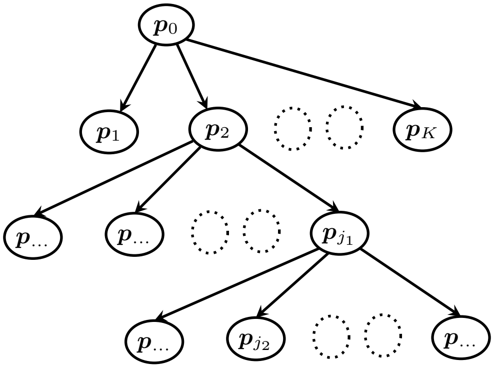
The basic Poisson-Dirichlet Process is a device for introducing infinite mixture models and for hierarchical modeling of discrete distributions. The basic form has as input a base probability distribution $H(\cdot)$ on a measurable space $\mathcal{X}$, and yields a discrete distribution on a finite or countably infinite subset of $\mathcal{X}$.

$$\sum_{k=1}^{\infty} p_k \delta_{X_k^*}(\cdot) \tag{1}$$

where $\boldsymbol{p} = (p_1, p_2, ...)$ is a probability vector so $0 \leq p_k \leq 1$ and $\sum_{k=1}^{\infty} p_k = 1$. Also, $\delta_{X_k^*}(\cdot)$ is a discrete measure concentrated at X_k^*. We assume the values $X_k^* \in \mathcal{X}$ are independently and identically distributed according to $H(\cdot)$, which is referred to as the *base distribution*. We also assume the base distribution is *discrete*, so $H(X) > 0$ for all samples $X \sim H(\cdot)$, although this is not generally the case for the PDP. The probability vector $\boldsymbol{p}$ follows a two-parameter Poisson-Dirichlet distribution [17]. A common definition for it is the "stick-breaking" model, as follows:

Definition 1 (Poisson-Dirichlet distribution). For $0 \leq a < 1$ and $b > -a$, suppose that a probability $P_{a,b}$ governs independent random variables V_k such that V_k has Beta$(1-a, b+k\,a)$ distribution. Let

$$p_1 \;=\; V_1, \quad p_k \;=\; (1-V_1)\cdots(1-V_{k-1})V_k \quad k \geq 2\ , \tag{2}$$



Probability vector hierarchy: This depicts, for instance, that vectors $\boldsymbol{p}_1$ to $\boldsymbol{p}_K$ should be similar to $\boldsymbol{p}_0$. So for the j_2-th node branching off node j_1, $\boldsymbol{p}_{j_2} \sim \text{PDP}(a_{j_1}, b_{j_1}, \boldsymbol{p}_{j_1})$. The root node $\boldsymbol{p}_0$ could be Dirichlet distributed if it is finite or could have a PDD distribution if infinite.

Fig. 1. Probability vector hierarchy

yielding $\boldsymbol{p} = (p_1, p_2, ...)$. Define the *Poisson-Dirichlet distribution* with parameters a, b, abbreviated PDD(a, b) to be the $P_{a,b}$ distribution of $\boldsymbol{p}$.

Note this does assume a particular ordering of the entries in $\boldsymbol{p}$. Here our a parameter is usually called the *discount parameter* in the literature, and b is called the *concentration parameter*. The DP is the special case where $a = 0$, and has some quite distinct properties such as slower convergence of the sum $\sum_{k=1}^{\infty} p_k$ to one. General results for the discrete case of the PDP are reviewed in [18].

A suitable definition of a *Poisson-Dirichlet process* is that it extends the Poisson-Dirichlet distribution using Formula (1), referred to as PDP$(a, b, H(\cdot))$. Thus the PDP is a functional on distributions: it takes as input a base distribution and yields as output a discrete distribution with a finite or countable set of possible values on the same domain.

The output distribution of a PDP can subsequently be used as a base distribution for another PDP, and so-forth, to create a hierarchy of distributions. This situation is depicted in the graphical model of Figure 1 where the distribution is over vectors $\boldsymbol{p}$ indexed by their positions in the hierarchy. Each vector represents a discrete probability distribution so is a suitable base distribution for the next level of PDPs. The hierarchical case for the DP is presented in [14], and the hierarchical and discrete case of the PDP in [19, 20]. The hierarchical PDP (HPDP) thus depends on the discrete distribution at the root node, and on the hyper-parameters a, b used at each node. This hierarchical occurrence of probability vectors could be a model in itself, as is the case for n-gram models of strings, or it could occur as part of some larger model, as is the case for the HDP-LDA model.

Intuitively, the HPDP structure can be well explained using the nested CRP mechanism, which has been widely used as a component of different topic models [21]. It goes as follows: a Chinese restaurant has an infinite number of tables, each of which has infinite seating capacity. Each table serves a dish, and multiple tables can serve the same dish. In the nested CRP, each restaurant is also linked to its parent restaurant and child restaurants in a tree-like structure. A newly

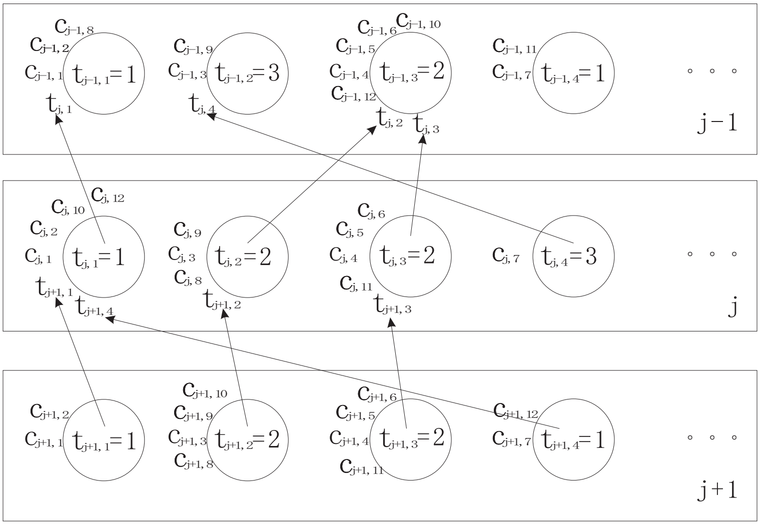


Fig. 2. Nested CRP representation of the HPDP, where rectangles correspond to restaurants, cycles correspond to tables, and C means customers

arrived customer can choose to sit at an active table (*i.e.*, a table which at least has one customer), or choose a new table. If a new table is chosen (*i.e.* activated), this table will be sent as a new customer to the corresponding parent restaurant, which means a table in any given restaurant reappears as a proxy customer [3] in its parent restaurant. This procedure is illustrated in Figure 2.

3 Related Methods

For the hierarchical PDP, the most popular MCMC algorithm is the Gibbs sampling method based on the Chinese restaurant representation [10, 14]. For instance, samplers proposed in [14], *e.g.*, the Chinese restaurant franchise sampler, the augmented Chinese restaurant franchise sampler, and the sampler for direct assignment. In the CRP representation, each restaurant is represented by a seating arrangement that contains the total number of customers, the total number of occupied tables, the customer-table association, the customer-dish association, and the table-dish association.

With the global probability measure marginalized out, the Chinese restaurant franchise sampler keeps track of the customer-table association (*i.e.*, recording table assignments of all customers), which results in extra storage space requirement. Its extension to the HPDP is a Gibbs sampler, so called "sampling for seating arrangements" by Teh [19]. Another sampler for HDP, termed "posterior sampling with an augmented representation", introduces an auxiliary variable to construct the global measure for its children DPs so that these DPs can be decoupled. A further extension of the augmented sampler gives the sampler for "direct assignment", in which each data point is directly assigned to one component, instead of sampling at which table it sits. This is the sampler used in Teh's implementation of HDP-LDA[22].

Recently, a new collapsed Gibbs sampling algorithm for the HPDP is proposed in [15, 18]. It sums out all the seating arrangements by introducing a constrained latent variable, called the table count that represents the number of tables serving the same dish, similar to the representation in the direct assignment sampler. Du *et al.* [16] have applied it to a first order Markov chain.

Except for the sampling based algorithms, there also exist variational based inference algorithms for the HPDP. For example, Wang *et al.* proposed a variational algorithm for the nested Chinese restaurant process [23], and recently proposed an online variational inference for the HDP [24]. As a compromise, Teh *et al.* developed a sampling-variational hybrid algorithm for HDP in [25].

In this paper, we propose a new table representation for the HPDP by introducing another auxiliary latent variable, called table indicator variable, to track which level the data (or customer) has contributed a table count (*i.e.* the creation of a new table) in the hierarchy. The aforementioned table count variable can be easily constructed from the table indicator variables by summation, which indicates the exchangeability of the proposed representation. To apply the new representation, we develop a block Gibbs sampling algorithm to jointly sample the dish and table indicator for each customer.

4 New Table Representation of the HPDP

In this section, we introduce a new table representation for the HPDP on top of the nested CRP configuration, as shown in Figure 2. Indeed, this new representation can be generalized to a framework for doing inference on the HPDP based models. We prove that the new representation allows full exchangeability by the joint posterior distribution, thus guarantees a correct Gibbs sampler to be developed. Although, in this work, we just study how the representation works on tree structure based graphical models, in which data items can be attached to any nodes in the tree, it can also be easily extended to arbitrary structures [20].

Note that, in the following elaboration, nodes and data items also correspond to restaurants and customers in the CRP, respectively; and dishes, as components in mixture models, correspond to, for instance, topics in probabilistic topic modeling[1]. Before further proceeding to the details of the new table representation, we first give three definitions that are used in the following sections.

Definition 2 (Node index j). *In the tree structure, each node (i.e., a restaurant) is indexed with an integer starting from 0 in a top-down and left to right manner. The root of the tree is indicated by $j = 0$. Based on this definition, we define a mapping $d : Z^+ \to Z^+$ which maps the node j to its level $d(j)$ in the tree, here Z^+ means non-negative integers.*

[1] We will use these terminologies interchangeably in this paper.

Definition 3 (Multiplicity t_k). *For the Chinese restaurant version of the PDP, assume the base distribution $H(\cdot)$ is discrete, it means the same dish can be served by multiple tables. The multiplicity t_k is thus defined as the number of tables serving the same dish k.*

Definition 4 (Table indicator u_l). *The table indicator u_l for each data item l (i.e., a customer) is an auxiliary latent variable which indicates up to which level in the tree l has contributed a table count (i.e. activated a new table). If $u_l = 0$, it means the data item l takes the responsibility of creating a new table in each node between the root and the current node.*

Here, what do we mean "table contribution" in Definition 4? In the nested CRP, as discussed in Section 2, if a newly arrived customer, denoted by l, chooses to sit at a new table, the table count in the corresponding restaurant, indexed by j, will be increased by one. Herein, opening up a new table is defined as the table contribution of this customer. Therefore, u_l is set to $d(j)$. Then, the new table is sent as a proxy customer to its parent restaurant, $pa(j)$. If the proxy customer again chooses to sit at a new table to contribute a table count, u_j will be set to $d\left(pa(j)\right)$. This procedure will proceed towards the root in a recursive way until there is no more table contribution. The introduction of table indicator variables is the core merit of the new representation.

In the tree structure, for each node j, every data item l in j is associated with a sampled value z_l, which can take K distinct values, *i.e.* K distinct dishes in the context of CRP; and a table indicator variable u_l is attached to l to trace its table contribution up towards the root. Thereby, taking advantage of table indicator variables, some statistics can be represented as follows:

$$n_{jk}^0 = \begin{cases} \sum_{l \in D(j)} \delta_{z_l=k}, & \text{for } D(j) \neq \emptyset \\ 0, & \text{others} \end{cases}, \quad t_{jk} = \sum_{j' \in T(j)} \sum_{l \in D(j')} \delta_{z_l=k} \delta_{u_l \leq d(j)}$$

$$n_{jk} = n_{jk}^0 + \sum_{j' \in C(j)} t_{j'k}, \quad T_j = \sum_k t_{jk}, \quad N_j = \sum_k n_{jk} \tag{3}$$

where n_{jk}^0 is the number of actual data points in j with $z_l = k$ ($k \in \{1, \ldots, K\}$), t_{jk} is the number of tables serving k, n_{jk} is the total number of data points (including those sent by the child nodes of j) with $z_l = k$, T_j is the total number of tables, and N_j is the total number of data points; and $D(j)$ is a set of data points attached to j, $C(j)$ is a set of child nodes of j and $T(j)$ is its closure, the set of nodes in the sub-tree rooted at j, Obviously, the multiplicity for each distinct value, *i.e.*, each dish, can be constructed from table indicator variables.

Lemma 1. *Given finite samples $z_1, z_2, \cdots, z_N$ from a Poisson-Dirichlet Process (i.e., $PDP(a, b, H)$), the joint probability of the samples and their multiplicities [18] $t_1, t_2, \cdots, t_K$ is given by:*

$$P_r(z_1, z_2, \cdots, z_N, t_1, \cdots, t_K) = \frac{(b|a)_T}{(b)_N} \prod_{k=1}^{K} \left(H(z_k^*)^{t_k} S_{t_k,a}^{n_k} \right) \tag{4}$$

where $S^N_{M,a}$ is the generalized Stirling number[2]*, $(x|y)_N$ denotes the Pochhammer symbol with increment y, and $T = \sum_{i=1}^K t_i$.*

Making use of Lemma 1 by [18], we can derive, based on the new table representation of the HPDP, the joint probability of samples $\boldsymbol{z}_{1:J}$ and table indicator variables $\boldsymbol{u}_{1:J}$ as

Theorem 1. *Given the base distribution H_0 for the root node, the joint posterior distribution of $\boldsymbol{z}_{1:J}$ and $\boldsymbol{u}_{1:J}$ for the HPDP in a tree structure is* [3]*:*

$$P_r(\boldsymbol{z}_{1:J}, \boldsymbol{u}_{1:J} \mid H_0) = \prod_{j \geq 0} \left(\frac{(b_j|a_j)_{T_j}}{(b_j)_{N_j}} \prod_k S^{n_{jk}}_{t_{jk},a_j} \frac{t_{jk}!(n_{jk} - t_{jk})!}{n_{jk}!} \right) . \tag{5}$$

Proof (sketch). Let us consider just one restaurant (*i.e.*, one node in the tree) at a time. The set $\boldsymbol{t} = \{t_1, \cdots, t_K\}$ indicates the table configuration, *i.e.*, the number of tables serving each dish; and the set $\boldsymbol{u} = \{u_1, \cdots, u_N\}$ as the table indicator configuration, *i.e.*, table indicators attached to each data item. Clearly, only one table configuration can be reconstructed from a table indicator configuration, as shown by Eq. (3). But given one table configuration, one can yield $\prod_k \frac{n_k!}{t_k!(n_k-t_k)!}$ possible table indicator configurations. Thereby, the joint posterior distribution of $\boldsymbol{z}$ and $\boldsymbol{t}$ is computed as

$$P_r(\boldsymbol{z}, \boldsymbol{t}) = \prod_k \frac{n_k!}{t_k!(n_k - t_k)!} P_r(\boldsymbol{z}, \boldsymbol{u}) \tag{6}$$

where $\boldsymbol{z} = (z_1, z_2, \cdots, z_N)$, t_k is the number of tables serving dish k, n_k is the number of customer eating dish k in the CRP metaphor. Note the chosen term says we can equally likely make just t_k of the n_k data items those that contribute tables. This formula lets us convert from the $(\boldsymbol{z}, \boldsymbol{t})$ representation of the PDP to $(\boldsymbol{z}, \boldsymbol{u})$ representation at a given node, assuming all its lower level nodes have already been converted. Thus we apply it recursively from the leaf nodes up. Combining this with Eq. (4) in Lemma 1 we can write down the joint probability of $\boldsymbol{z}$ and $\boldsymbol{u}$ for the tree structure as in Eq. (5).

Corollary 1. *The joint posterior distribution of $\boldsymbol{z}_{1:J}$ and $\boldsymbol{u}_{1:J}$ is exchangeable in the pairs (z_j, u_j).*

Proof (sketch). Follows by inspection of the posterior and that the statistics used are all sums over data items.

[2] A generalized Stirling number is given by the linear recursion [15, 16, 18, 19] as $S^{N+1}_{M,a} = S^N_{M-1,a} + (N - Ma)S^N_{M,a}$, for $M \leq N$. It is 0 otherwise and $S^N_{0,a} = \delta_{N,0}$. These numbers rapidly become very large so computation needs to be done in log space using a logarithmic addition.

[3] Note that $t_{0k} \leq 1$ since this node is the PDD defined in definition 1.

5 Block Gibbs Sampling Algorithm

In this section, we elaborate on our block Gibbs sampling algorithm for the new table representation, and show, as an example, how it can be applied to the HDP-LDA model.

5.1 Block Gibbs Sampler

In Gibbs sampling, the new state is reached by sequentially sampling all variables from their distribution conditioned on the previous state, *i.e.*, the current values of all other variables and the data. Unlike the collapsed Gibbs sampling algorithms proposed in [15, 16, 20], all of which divide the Gibbs sampler into two parts, *i.e.* sampling z and t separately, instead, we develop for the new table representation a block Gibbs sampling algorithm that jointly samples the variable z and its table indicator variable u based on Theorem 1. The full joint conditional distribution $P_r(z_l, u_l \mid \boldsymbol{z}_{1:J} - z_l, \boldsymbol{u}_{1:J} - u_l)$ can be obtained by a probabilistic argument or by cancellation of terms in Eq. (5).

While doing the joint sampling, the old value of table indicator u_l first needs to be removed simultaneously along with the old value of z_l in the probabilistic argument. The challenge here is to deal with the coupling between u_l and z_l, and that between u_l and data points. Note that for some cases, the tables created by data item l up to the level u_l cannot be removed since they are being shared with other data items, discussed in Subsection 5.2. In such a case, we need to skip the table operation for the datum l and proceed to other data items until the tables linked to l can be removed. More specifically, the sampling procedure goes as follows:

First, define $path(l)$ containing a set of nodes to be a path from the root to the leaf to which the data item has table contribution; T_j, N_j, t_{jk}, n_{jk} are defined in Section 4, $T'_j, N'_j, t'_{jk}, n'_{jk}$ are the corresponding values after removing the datum l, and $T''_j, N''_j, t''_{jk}, n''_{jk}$ are those after adding l back again. To sample (z_l, u_l), we need to consider all nodes $j \in path(l)$ because removing l would probably change table configurations for these nodes. It is easy to show that these variables are subject to the following relations:

For all node $j \in path(l)$, after removing l (note l only belongs to one node in the path), we have

$$T'_j = \begin{cases} T_j, & \text{if } u_l > d(j) \\ T_j - 1, & \text{if } u_l \le d(j) \end{cases}, \quad t'_{jk} = \begin{cases} t_{jk}, & \text{if } u_l > d(j) \\ t_{jk} - 1, & \text{if } u_l \le d(j) \,\&\, z_l = k \end{cases}$$

$$N'_j = \begin{cases} N_j, & \text{if } l \notin D(j) \,\&\, u_l > d(j) + 1 \\ N_j - 1, & \text{if } l \notin D(j) \,\&\, u_l \le d(j) + 1 \text{ or } l \in D(j) \end{cases}$$

$$n'_{jk} = \begin{cases} n_{jk}, & \text{if } l \notin D(j) \,\&\, u_l > d(j) + 1 \\ n_{jk} - 1, & \text{if } \ (l \notin D(j) \,\&\, u_l \le d(j) + 1 \text{ or } l \in D(j)) \,\&\, z_l = k \end{cases} . \tag{7}$$

After adding l, u_l:

$$T''_j = \begin{cases} T'_j, & \text{if } u_l > d(j) \\ T'_j + 1, & \text{if } u_l \leq d(j) \end{cases}, \quad t''_{jk} = \begin{cases} t'_{jk}, & \text{if } u_l > d(j) \\ t'_{jk} + 1, & \text{if } u_l \leq d(j) \,\&\, z_l = k \end{cases}$$

$$N''_j = \begin{cases} N'_j, & \text{if } l \notin D(j) \,\&\, u_l > d(j)+1 \\ N'_j + 1, & \text{if } l \notin D(j) \,\&\, u_l \leq d(j)+1 \text{ or } l \in D(j) \end{cases}$$

$$n''_{jk} = \begin{cases} n'_{jk}, & \text{if } l \notin D(j) \,\&\, u_l > d(j)+1 \\ n'_{jk} + 1, & \text{if } (l \notin D(j) \,\&\, u_l \leq d(j)+1 \text{ or } l \in D(j)) \,\&\, z_l = k \end{cases} . \tag{8}$$

With respect to the above analysis, the full joint conditional probability for z_l and u_l is:

$$\begin{aligned} & P_r(z_l, u_l \,|\, \boldsymbol{z}_{1:J} - z_l, \boldsymbol{u}_{1:J} - u_l, H_0) \\ & = \prod_{j \in path(l)} \frac{(b_j + a_j T'_j)^{\delta_{(T''_j \neq T'_j)}}}{(b_j + N'_j)^{\delta_{(N''_j \neq N'_j)}}} \left(\frac{S^{n''_{jk}}_{t''_{jk}, a_j}}{S^{n'_{jk}}_{t'_{jk}, a_j}} \right)^{\delta_{n''_{jk} \neq n'_{jk} \| t''_{jk} \neq t'_{jk}}} \\ & \quad \frac{(t''_{jk})^{\delta_{t''_{jk} \neq t'_{jk}}} (n''_{jk} - t''_{jk})^{\delta_{n''_{jk} - t''_{jk} \neq n'_{jk} - t'_{jk}}}}{(n''_{jk})^{\delta_{n''_{jk} \neq n'_{jk}}}} . \end{aligned} \tag{9}$$

5.2 Constraint Analysis

As discussed in the previous section, when we add or remove a data item from the tree, the statistics associated with this data item cannot be changed unless some conditions are satisfied. For removing a data item l from the tree, the related statistics can be changed if one of the following conditions holds:

1. Data l has no table contribution, *i.e.*, $u_l = L$, or
2. No other data points share the tables created by data l, *i.e.*, $\forall_{j \in path(l)}(d(j) \geq u_l \implies n_{jk} = 1)$, or
3. There are other tables created by other data points sharing the same dishes with those created by data l in the same nodes, *i.e.*, $\forall_{j \in path(l)}(d(j) \geq u_l \implies t_{jk} > 1)$.

When l is added back to the tree, its table indicator cannot always take any value from 0 to L, they should be constrained in some range, *i.e.*, if l is added to menu item k with current table count $t_k = 0$, l will created a new table and contribute one table count to the current node (*i.e.*, $t_k + 1$), and u_l will be set to $d(j)$ such that $u_l < L$. Thus, the value of u_l should be in the following interval:

$$u_l \in [u_l^{min}, u_l^{max}]$$

where u_l^{min} denotes the minimum value of the table indicator for j, and u_l^{max} the maximum value. These can be shown as

$$u_l^{max} = \begin{cases} \min\left\{d(j) : j \in path(l), t_{jk} = 0\right\}, \text{if } \exists j, t_{jk} = 0 \\ L, \qquad\qquad\qquad\qquad\qquad\qquad\quad \text{if } \forall j, t_{jk} > 0 \end{cases}$$

$$u_l^{min} = \begin{cases} 0, \text{if } t_{0k} = 0 \\ 1, \text{others} \end{cases} . \tag{10}$$

5.3 Application to the HDP-LDA Model

In this section, we apply our proposed table representation and block Gibbs sampler to the hierarchical Dirichlet process variant of latent Dirichlet allocation (HDP-LDA)[14] in topic modeling. In the HDP-LDA model, each document corresponds to a restaurant in the Chinese restaurant representation, or a DP in the HDP. All the restaurants share the same parent, that is to say all the DPs share the same global probability measure drawn from another DP.

To develop the sampling formulas for the HDP-LDA model, we need to incorporate the prior distribution for the topics in the joint distribution shown by Eq. (5). The prior distribution of the topic-word matrix adopted in our work is the Dirichlet distribution. Thus, given K topics, W words in vocabulary, and a topic-word matrix $\boldsymbol{\Phi}_{K\times W}$ which has a Dirichlet distributed prior $\boldsymbol{\gamma}$, the joint distribution of $\boldsymbol{z}_{1:J}, \boldsymbol{u}_{1:J}$ can be derived by incorporating the prior in Eq. (5) and integrating out all ϕ_k as

$$\begin{aligned} &P_r(\boldsymbol{z}_{1:J}, \boldsymbol{u}_{1:J}) \\ &= \prod_{j\geq 0} \left(\frac{(b_j|a_j)_{T_j}}{(b_j)_{N_j}} \prod_k S^{n_{jk}}_{t_{jk},a_j} \frac{t_{jk}!(n_{jk}-t_{jk})!}{n_{jk}!} \right) \prod_k \left(\frac{Beta_W(\boldsymbol{\gamma}+\boldsymbol{M}_k)}{Beta_W(\boldsymbol{\gamma})} \right) \end{aligned} \tag{11}$$

where $\boldsymbol{M}_k$ denotes the number of words attached to topic k in the document collection, $\text{Beta}_W(\gamma)$ is W dimensional beta function that normalizes the Dirichlet. Specifically, in the case of HDP-LDA, we have $a_j = 0, \forall j$, and documents are indexed by $j \in [1, D]$.

Now, beginning with the joint distribution, Eq. (11), using the chain rule, we obtain the full joint conditional distribution according to the statistics before/after removing j and after adding j as follows:

1. If $\forall j', t'_{j'k} = 0$,

$$P_r(z_l = k_{new}, u_l = u \,|\, \boldsymbol{z}_{1:J} - z_l, \boldsymbol{u}_{1:J} - u_l) \propto \frac{b_0 b_1}{b_0 + \sum_k Tt[k]} \frac{\gamma_l + M_{kl}}{\sum_{l'}(\gamma_{l'} + M_{kl'})} \tag{12}$$

2. If $t'_{jk} \neq 0, t'_{0k} \neq 0$,

$$\begin{aligned} &P_r(z_l = k, u_l = u \,|\, \boldsymbol{z}_{1:J} - z_l, \boldsymbol{u}_{1:J} - u_l) \\ &\propto \frac{S^{n''_{jk}}_{t''_{jk},0}}{S^{n'_{jk}}_{t'_{jk},0}} \frac{(t''_{jk})^{\delta_{t''_{jk}\neq t'_{jk}}} (n''_{jk} - t''_{jk})^{\delta_{n''_{jk}-t''_{jk}\neq n'_{jk}-t'_{jk}}}}{(n''_{jk})^{\delta_{n''_{jk}\neq n'_{jk}}}} \frac{\gamma_l + M_{kl}}{\sum_{l'}(\gamma_{l'} + M_{kl'})} \end{aligned} \tag{13}$$

3. If $t'_{jk} = 0, t'_{0k} \neq 0$,

$$\begin{aligned} & P_r(z_l = k, u_l = u \,|\, \boldsymbol{z}_{1:J} - z_l, \boldsymbol{u}_{1:J} - u_l) \\ & \propto \frac{b_1 Tt[k]^2}{(Tt[k]+1)(\sum_k Tt[k] + b_0)} \frac{\gamma_l + M_{kl}}{\sum_{l'}(\gamma_{l'} + M_{kl'})} \end{aligned} \tag{14}$$

where $Tt[k]$ denotes the number of tables serving dish k (*i.e.*, topic k), M_{kl} indicates the total number of words l assigned to k in the document collection.

Note that implementing the block sampler requires keeping track of the table indicators for each data item. This can in fact be avoided in the same way that the Sampling for Seating Arrangements approach [19, 26] works, the values can be randomly reconstructed bottom up as needed. A single u_l can be sampled this way, and the remaining $\boldsymbol{u}_{1:J} - u_l$ do not need to be sampled since only their statistics are needed and these can be reconstructed from $\boldsymbol{t}$ and u_l. Thus our implementation records the $\boldsymbol{t}$ but not $\boldsymbol{u}_{1:J}$.

6 Experiments

We compared the proposed algorithm with Teh *et al.*'s [14] "posterior sampling by direct assignment" sampler[4] as well as Buntine and Hutter's collapsed sampler [18] on five datasets, namely, *Health* dataset, *Person* dataset, *Obama* dataset, *NIPS* dataset and *Enron* dataset. All three algorithms are implemented in C, and run on a desktop with Intel(R) Core(TM) Qaud CPU (2.4GHz), although our code is not multi-threaded.

The *Obama* dataset came from a collection of 7M Blogs (from ICWSM 2009, posts from Aug-Sep 2008) by issuing the query "obama debate" under Lucene. We performed fairly standard tokenization, created a vocabulary of terms that occurred at least five times after excluding stopwords, and then built the bag of words. The *Health* data set is similar except the source is 1M News articles (LDC Gigaword) using the query "health medicine insurance" and words needed to occur at least twice. The *Person* dataset has the source of 805k News articles (Reuters RCV1) using the query "person" and using words that occurred at least four times. The *Enron* and *NIPS* datasets have been obtained as preprocessed bagged data from UCI where the data has been used in several papers. We list some statistics of these datasets in Table 1.

Table 1. Statistics of the five datasets

	Health	*Person*	*Obama*	*NIPS*	*Enron*
# words	1,119,678	1,656,574	1,382,667	1,932,365	6,412,172
# documents	1,655	8,616	9,295	1,500	39,861
vocabulary size	12,863	32,946	18,138	12,419	28,102

[4] The "posterior sampling by direct assignment" is preferred than the other two samplers in [14], due to its straightforward bookkeeping, as suggested by Teh *et al.*

6.1 Experiment Setup and Evaluation Criteria

The algorithms tested are: our proposed block Gibbs sampler for HDP-LDA, **S**ampling by **T**able **C**onfigurations, denoted as STC, Teh *et al.*'s "**S**ampling by **D**irect **A**ssignment" algorithm[14], denoted as SDA, and the **C**ollapsed Gibbs **T**able **S**ampler by Buntine *et al.*[18], denoted as CTS, and finally, a variance of the proposed STC by initializing word topics with SDA and sampling tables for each document using STC, denoted as SDA+STC. The reason for using the fourth algorithm is to isolate the impact of the new sampler.

Coming to the evaluation criteria for topic models, there are many different evaluation methods such as importance sampling methods, Harmonic mean method, "left-to-right" algorithm, *etc.*, see [27, 28] for a complete survey. In this paper, we adopt the "left-to-right" algorithm to calculate the test perplexities because it is unbiased [27]. This algorithm calculates the perplexity over words following a "left-to-right" manner for each document, which is defined as[27, 28]:

$$P_r(w|\Phi, \alpha m) = \prod_n P_r(w_n|w_{<n}, \Phi, \alpha m) \tag{15}$$

where w is all the words in the documents, Φ is the topic distribution matrix, α is the concentration parameter of the Dirichlet prior over topics, and m is its base measure. The perplexity is computed in the log space. Since table counts are also latent variables in the formulation of perplexity, we force all table counts to be less than or equal to one so that the unbiased method can be applied directly. This condition has been observed to hold generally when the PDP hyperparameters are well optimized/fit.

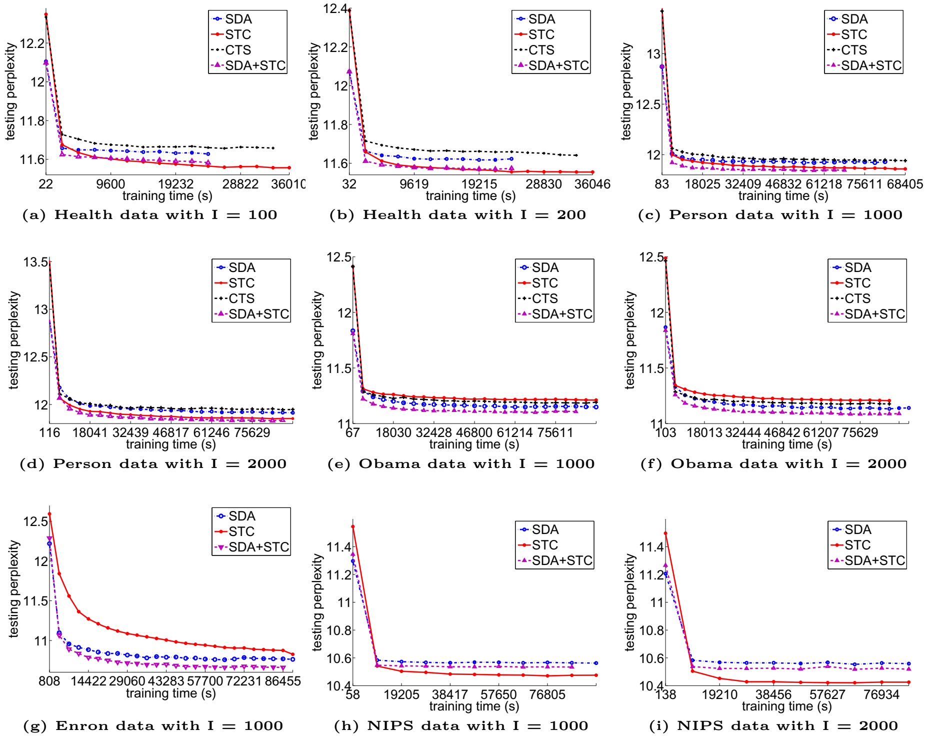
6.2 Parameter Setting

While there are many parameters in the proposed model, *e.g.*, the two parameters of the PDP a_i, b_i for each document, for simplicity, we set all the b_i's in the same level of the tree structure to the same value, and optimize it by sampling, meanwhile we set all a_i's to 0 since we are testing the HDP model. More specifically, we follow Teh *et al.*[1], to sample b_i by introducing Beta distributed auxiliary variables for each document.

For other parameters, we need to set the number of test documents for each dataset. In the experiments, we set these approximately to 5% of the total size of dataset. Specifically, the number of test documents for Health dataset is 250/1655, and 500/8616 for Person dataset, 500/9259 for Obama dataset, 50/1500 for NIPS dataset, and 500/39861 for Enron dataset. Here, x/y means x out of total y documents are used for testing. Moreover, we set the symmetric Dirichlet prior of word distributions over topics (this is Dirichlet prior for the topic matrix Φ in the LDA[13] case) to 0.01, a default value used in most of LDA experiments.

Table 2. Test log_2(perplexities) on the five datasets

Dataset	*Health*		*Person*		*Obama*	
	$I = 100$	$I = 200$	$I = 1000$	$I = 2000$	$I = 1000$	$I = 2000$
SDA	11.628281	11.619546	11.930657	11.904425	11.144188	11.134732
CTS	11.655493	11.636743	11.940532	11.947740	11.191377	11.174327
SDA+STC	**11.582969**	**11.573457**	**11.844319**	**11.829628**	**11.094079**	**11.090389**
STC	**11.547999**	**11.551453**	**11.858719**	**11.852253**	11.210295	11.201241
Dataset	*Enron*		*NIPS*			
	$I = 500$	$I = 1000$	$I = 1000$	$I = 2000$		
SDA	10.847454	10.768568	10.564221	10.558330		
SDA+STC	**10.768568**	**10.659724**	**10.534148**	**10.518792**		
STC	10.899923	10.810127	**10.474467**	**10.425393**		



(a) Health data with I = 100

(b) Health data with I = 200

(c) Person data with I = 1000

(d) Person data with I = 2000

(e) Obama data with I = 1000

(f) Obama data with I = 2000

(g) Enron data with I = 1000

(h) NIPS data with I = 1000

(i) NIPS data with I = 2000

Fig. 3. Test $\log_2$(perplexities) evolved with training time, I means initial number of topics

6.3 Perplexities

We first give the experimental results for the four algorithms on the five datasets in term of testing perplexity using the "left-to-right" algorithm [27, 28]. We also need to set the initial number of topics for each of these algorithms though the final values can be sampled by these algorithms. Generally, to accelerate the convergent speed, we initialized more topics to large datasets than those to small datasets. Specifically, we initialized Health dataset with 100 and 200 topics, Enron dataset with 500 and 1000 topics[5], and other three datasets with 1000 and 2000 topics, respectively. Furthermore, we used 2000 major cycles to burn-in[6]. Table 2 summarizes the test perplexities for these algorithms.

From Table 2, we can see that the proposed fully exchangeable block Gibbs sampler STC obtains significantly better results than the "sampling by direct assignment" sampler does in most cases[7], while using SDA to burn in, and sampling tables with the proposed corrected sampler, SDA+STC obtains consistently better results than SDA. One interesting observation is that the collapsed Gibbs sampler does not perform as well as expected, and we expect this is due to poor mixing of the Gibbs sampler since the table counts are updated separately from the topics.

6.4 Convergence Speed

Although our proposed sampler looks much more complex, the convergence speed is almost as fast as Teh *et al.*'s and actually converges better. To verify this, we set up another set of experiments. We started all the algorithms with a clock to record the total training time, and randomly initialized these samplers. When the training time has elapsed for an amount of time, say 1 hour for large datasets and 40 minutes for small datasets, we calculated the testing perplexities on the corresponding testing datasets using the current statistics, and recorded the perplexities. In order to do a fair comparison, all codes for these algorithms have been carefully implemented to reach their optimal speed. We observed that although the running time for each Gibbs cycle of our algorithm is slightly longer than that of Teh *et al.*'s algorithm, our algorithm can still converge very fast in most cases. Figure 3 plots the trends of testing perplexities on each datasets corresponding to the training time. We can see from the figures that the proposed algorithm converges well except on the largest dataset Enron. Subsequent experiments show this could be overcome with better initialization.

7 Conclusion

In this paper, we have proposed a new table representation that inherits the full exchangeability from the two-parameter Poisson-Dirichlet processes or Dirichlet Processes to their hierarchical extensions, and developed a block Gibbs sampling algorithm for doing inference on this representation. Meanwhile, we have

[5] This dataset is too large to initialize with large number of initial topics.

[6] For some large datasets, a 2000 burn-in is impractical, thus we set a maximum burn-in time for 24 hours.

[7] There is a case, *i.e.*, *Obama* dataset, that STC is not as good as SDA, this may be due to the initialization.

applied the proposed algorithm to the HDP-LDA model, and used the block Gibbs sampler to jointly sample the topic and table indicator configuration for each word.

Experimental results showed that with proper initializations and sampling only tables, SDA+STC can yield consistently better results than "sampling by direct assignment" algorithm SDA does in term of testing perplexity; and the block Gibbs sampler STC with full exchangeability can always outperform SDA. Furthermore, we have demonstrated that though the proposed model is more complicated than the original one, its convergence speed is always faster than the "sampling by direct assignment" algorithm's. Interestingly, an earlier alternative collapsed Gibbs sampler performed poorly, and we expect this is because STC allows better mixing of the HPDP/HDP parts of the model with the other parts of the model.

We claim that our new representation and the performance improvements should extend to many of the other Gibbs algorithms now in use for different models embedding the HPDP or HDP. Thus our methods can be applied quite broadly within the non-parameteric Bayesian community. The superiority of our method over the various CRP-based approaches mentioned in Section 3 and our earlier collapsed sampler are:

- The introduction of the table indicator variable guarantees full exchangeability. This is important to eliminate sequential effects from the Gibbs sampler.
- Tracking the table contribution can reduce the information loss that may result from summing out all the seating arrangements. This makes the Gibbs sampler more rapidly mixing.

Acknowledgments. 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 Center of Excellence program.

References

1. Teh, Y.W.: A hierarchical Bayesian language model based on Pitman-Yor processes. In: ACL 2006, pp. 985–992 (2006)
2. Goldwater, S., Griffiths, T., Johnson, M.: Interpolating between types and tokens by estimating power-law generators. In: NIPS 2006, pp. 459–466 (2006)
3. Mochihashi, D., Sumita, E.: The infinite Markov model. In: NIPS 2008, pp. 1017–1024 (2008)
4. Johnson, M., Griffiths, T., Goldwater, S.: Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. In: NIPS 2007, pp. 641–648 (2007)
5. Wallach, H., Sutton, C., McCallum, A.: Bayesian modeling of dependency trees using hierarchical Pitman-Yor priors. In: Proceedings of the Workshop on Prior Knowledge for Text and Language (in Conjunction with ICML/UAI/COLT), pp. 15–20 (2008)
6. Wood, F., Archambeau, C., Gasthaus, J., James, L., Teh, Y.: A stochastic memoizer for sequence data. In: ICML 2009, pp. 119–116 (2009)

7. Rasmussen, C.: The infinite Gaussian mixture model. In: NIPS 2000, pp. 554–560 (2000)
8. Pruteanu-Malinici, I., Ren, L., Paisley, J., Wang, E., Carin, L.: Hierarchical Bayesian modeling of topics in time-stamped documents. TPAMI 32, 996–1011 (2010)
9. Xu, Z., Tresp, V., Yu, K., Kriegel, H.P.: Infinite hidden relational models. In: UAI 2006, pp. 544–551 (2006)
10. Teh, Y.W., Jordan, M.I.: Hierarchical Bayesian nonparametric models with applications. In: Bayesian Nonparametrics: Principles and Practice (2010)
11. Ishwaran, H., James, L.: Gibbs sampling methods for stick-breaking priors. Journal of ASA 96, 161–173 (2001)
12. Buntine, W., Jakulin, A.: Discrete components analysis. In: Subspace, Latent Structure and Feature Selection Techniques (2006)
13. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (2003)
14. Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.M.: Hierarchical Dirichlet processes. Journal of the ASA 101, 1566–1581 (2006)
15. Du, L., Buntine, W., Jin, H.: A segmented topic model based on the two-parameter Poisson-Dirichlet process. Mach. Learn. 81, 5–19 (2010)
16. Du, L., Buntine, W., Jin, H.: Sequential latent Dirichlet allocation: Discover underlying topic structures within a document. In: ICDM 2010, pp. 148–157 (2010)
17. Pitman, J., Yor, M.: The two-parameter Poisson-Diriclet distribution derived from a stable subordinator. Annals Prob. 25, 855–900 (1997)
18. Buntine, W., Hutter, M.: A Bayesian review of the Poisson-Dirichlet process. Technical Report arXiv:1007.0296, NICTA and ANU, Australia (2010)
19. Teh, Y.: A Bayesian interpretation of interpolated Kneser-Ney. Technical Report TRA2/06, School of Computing, National University of Singapore (2006)
20. Buntine, W., Du, L., Nurmi, P.: Bayesian networks on Dirichlet distributed vectors. In: PGM 2010, pp. 33–40 (2010)
21. Blei, D.M., Griffiths, T.L., Jordan, M.I.: The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. J. ACM 57, 1–30 (2010)
22. Teh, Y.: Nonparametric Bayesian mixture models - release 2.1. Technical Report University College London (2004), `http://www.gatsby.ucl.ac.uk/~ywteh/research/software.html`
23. Wang, C., Blei, D.: Variational inference for the nested Chinese restaurant process. In: NIPS 2009, pp. 1990–1998 (2009)
24. Wang, C., Paisley, J., Blei, D.: Online variational inference for the hierarchical Dirichlet process. In: AISTATS 2011 (2011)
25. Teh, Y., Kurihara, K., Welling, M.: Collapsed variational inference for HDP. In: NIPS 2007 (2007)
26. Blunsom, P., Cohn, T., Goldwater, S., Johnson, M.: A note on the implementation of hierarchical Dirichlet processes. In: ACL 2009, pp. 337–340 (2009)
27. Buntine, W.: Estimating likelihoods for topic models. In: Zhou, Z.-H., Washio, T. (eds.) ACML 2009. LNCS, vol. 5828, pp. 51–64. Springer, Heidelberg (2009)
28. Wallach, H., Murray, I., Salakhutdinov, R., Mimno, D.: Evaluation methods for topic models. In: ICML 2009, pp. 672–679 (2009)