# HIERARCHICAL TOPIC MODELS AND THE NESTED CHINESE RESTAURANT PROCESS

**Jiayi Ding**
Department of Statistical Science
Duke University
jiayi.ding@duke.edu

**Yangfan Ren**
Department of Statistical Science
Duke University
yangfan.ren@duke.edu

**Chudi Zhong**
Department of Statistical Science
Duke University
chudi.zhong@duke.edu

April 29, 2019

## ABSTRACT

Latent Dirichlet allocation (LDA) is a generative probabilistic model for collections of discrete data. It is also a three-level hierarchical Bayesian model which contains words, topics, and documents. In the text modeling, it considers that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. LDA specifies that a word has a multinomial distribution conditioned on the topic and a topic also follows a multinomial distribution given $\theta$ which is determined to be a Dirichlet distribution. However, topics are not always independent with each other and new entities and structures are brought by the growth of data sets. LDA is not flexible enough confronted with these conditions and infeasible to provide a hierarchical structure of topics. Therefore, the nested Chinese restaurant process (nCRP) which allows arbitrarily large branching factors and readily accommodates growing data collection is adopted as a nonparametric prior. This prior combined with a likelihood that is based on a hierarchical variant of LDA constructs a hierarchical topic model.

In this project, we implement hLDA and the nested Chinese restaurant process in python, optimize it by cython, and apply it to simulated and real data to compare the performance and efficiency.

***K*eywords** hierarchical Bayesian model · latent Dirichlet allocation · nested Chinese restaurant process · cython

## 1 Background

We implement hLDA referring to the paper *Hierarchical Topic Models and the Nested Chinese Restaurant Process* published in 2004[1] by David M. Blei, Michael I. Jordan, Thomas L. Griffiths, and Joshua B. Tenenbaum. The paper extends LDA proposed in 2003 to a hierarchical topic model with a prior via the nCRP.

Complex probabilistic models are increasingly prevalent in various domains since data sets often grow over time and bring new entities and new structures. In the topic modeling, given a collection of “documents”, each of which contains a set of “words”, we want to find common usage “topics” in the documents and to organize these topics into a hierarchy. The paper incorporates the Bayesian approach with the prior constructed by the nested Chinese restaurant process and likelihood based on a hierarchical variant of latent Dirichlet allocation to adapt to the accumulation of data.

As for hLDA method, each node in the hierarchy is associated with a topic, where a topic is a distribution across words. A document is generated by choosing a path from the root to a leaf, repeatedly sampling topics along that path, and sampling the words from the selected topics. This structure can capture the breadth of usage of topics across the corpus, reflecting underlying syntactic and semantic notions of generality and specificity. Compared with other proposed methods in this area, this approach takes the advantage of lack the assumption that the distributions associated with parent nodes and their descendants.

## 2 Algorithm

We describe the algorithm in four subsections: the nested Chinese restaurant process (nCRP), latent Dirichlet allocation (LDA), hLDA, and approximate inference by Gibbs sampling.

### 2.1 The nested Chinese restaurant process

The *Chinese restaurant process (CRP)* is a distribution on partitions of integers. Assume that a Chinese restaurant has an infinite number of tables. The first customer sits at the first table, and the subsequent customer sits at a table with the following distribution:

$$\begin{aligned} \Pr(occupiedtablei|previouscustomers) &= \frac{m_i}{\gamma + m - 1} \\ \Pr(nextunoccupiedtable|previouscustomers) &= \frac{\gamma}{\gamma + m - 1} \end{aligned}$$

where $m_i$ is the number of the previous customers in the table $i$, $m - 1$ is the number of customers in the restaurant when $m$th customer arrived, and $\gamma$ is a parameter. After the $M$ customers sit down, the seating plan gives the partition of $M$ items.

A *nested Chinese restaurant process (nCRP)* extends CRP to a hierarchical version. Suppose that there is an infinite number of the infinite-table Chinese restaurant in a city. One restaurant is the root restaurant and each of the infinite tables has a card directing to another restaurant. This structure can be represented by an infinitely-branched tree.

Consider the tourist case. A tourist arrives in the city and enters the root Chinese restaurant and selects a table following the distribution mentioned above. On the second evening, he goes to the restaurant identified on the table last night and chooses another table. Let $L$ be the number of days that a tourist visits the city. The tourist goes to $L$ restaurant, thereby constituting a path from the root to a restaurant at the $L$th level in the infinite tree. If $M$ tourists take $L$-day trips, the collection of paths shows a particular $L$-level subtree of the infinite tree.

Similar to the standard CRP which can be used to express uncertainty about a possible number of components, the nested CRP can be used to describe uncertainty about possible $L$-level trees.

### 2.2 Latent Dirichlet allocation

LDA is a generative probabilistic model for collections of discrete data. In the context of text modeling, the data set composed of a *corpus* of documents, and each *document* is a collection of words, where *word* is an item in a *vocabulary*. We define the following terms for simplicity:

- A *word* is the basic unit of discrete data and we use $w$ to represent each word.
- A *document* is a sequence of $N$ words denoted by $w = (w_1, w_2, \ldots, w_N)$, where $w_n$ is the $n$th word in the sequence.
- A *corpus* is a collection of $M$ documents represented by $D = \{w_1, w_2 \ldots, w_M\}$

The basic idea of LDA is that documents are represented as a random mixtures over latent topics and each topic is determined by a distribution of words. We use the following process to explain the generative process for each $w$ in a corpus $D$:

1. choose $N \sim Poisson(\xi)$
2. choose $\theta \sim Dirichlet(\alpha)$
3. For each of N words $w_n$:
   (a) choose topic $z_n \sim Multinomial(\theta)$
   (b) choose word $w_n \sim Multinomial(z_n, \beta)$

The LDA model is represented in Figure 1, which clearly visualizes three levels. The parameters $\alpha$ and $\beta$ are corpus level parameters and $\theta$ are document-level variables. The variables $z$ and $w$ are word-level variables and sampled once for each word in each document. With the three-level structure, LDA allows one document to be associated with multiple topics.

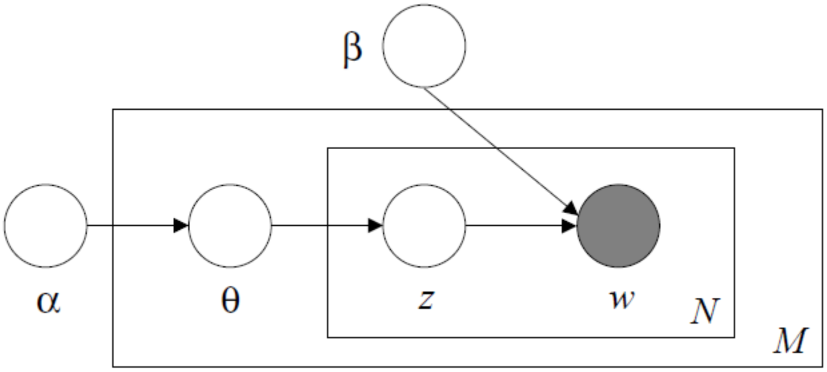


Figure 1: Graphic representation of LDA model. The outer box represents documents and the inner box represents topics and words within a document. (Source: D. Blei, et al., 2003)

### 2.3 A hierarchical topic model

A hierarchical topic model extends the LDA model in which the topic lies in a hierarchy. Suppose there is an $L$-level tree and each node is associated with a topic, the following process illustrates how documents are generated:

1. choose a path from the root of the tree to a leaf
2. choose $\theta \sim Dirichlet(\alpha)$
3. generate words in the document from a mixture topics along the selected path

Now we relax the assumption of a fixed $L$-level tree and use the nested CRP to place a prior on possible trees. The generative process of the hierarchical topic model is explained below:

1. let $c_1$ be the root node
2. for each level $l \in \{2, ..., L\}$:
   (a) choose a table from restaurant $c_{l-1}$ using the equation mentioned in 2.1
   (b) let $c_l$ be the next restaurant referred to by that table
3. for each word $n \in \{1, ..., N\}$:
   (a) choose $z \in \{1, ..., L\}$ from $multinomial(\theta)$
   (b) choose $w_n$ from the topic associated with restaurant $c_z$

Figure 2 displays the hLDA model, where node labeled $T$ refers to a collection of an infinite number of $L$-level paths drawn from a nested CRP.

### 2.4 Approximate Inference by Gibbs sampling

Gibbs sampling algorithm samples from the posterior nested CRP and corresponding topics in the hLDA model. This method provides the information about the parameter space after we get the data $w_{m,n}$, the $n$th word in the $m$th document. In this case, parameters that we are interested in are:

- $c_{m,l}$: the $l$th topic in document $m$
- $z_{m,n}$: topic that the $n$th word in the $m$th document is assigned to

First of all, we sample $z_{m,n}$:

$$\Pr(z_i = j | z_{-i}, w) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + w\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha} \tag{1}$$

where

- $z_{-i}$: assignment of all $z_k$ such that $k \neq i$

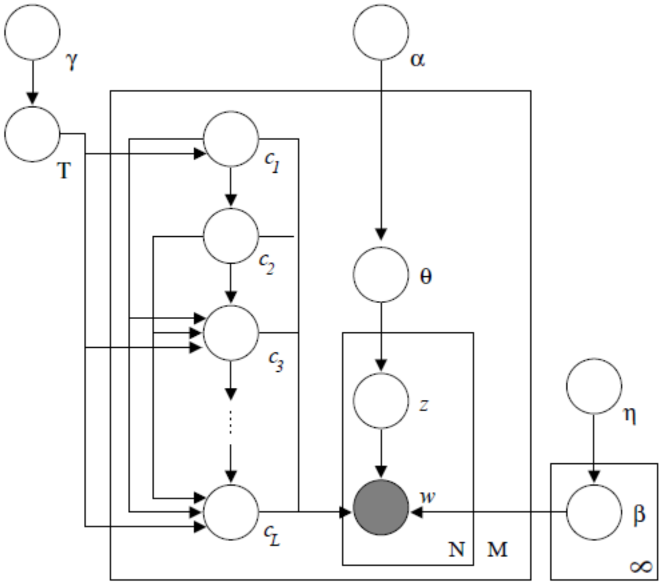


Figure 2: Graphic representation of hLDA model with the nested CRP prior. (Source: T. Griffiths, et al., 2004)

- $n^{(w_i)}_{-i,j}$: number of words assigned to topic $j$ that are same as $\omega_i$
- $n^{(\cdot)}_{-i,j}$: total number of words assigned to topic $j$
- $n^{(d_i)}_{-i,\cdot}$: total number of words in document $d_i$
- $n^{(d_i)}_{-i,j}$: number of words from document $d_i$ assigned to topic $j$

In order to get the distribution of $c_m$, we first derive the likelihood distribution $w_m|c, w_{-m}, z$:

$$\Pr(w_m|c, w_{-m}, z) = \prod_{l=1}^{L}\Big(\frac{\Gamma(n^{(\cdot)}_{c_{m,l},-m} + W\eta)}{\prod_w \Gamma(n^{(w)}_{c_{m,l},-m} + \eta)} \frac{\prod_w \Gamma(n^{(w)}_{c_{m,l},-m} + n^{(w)}_{c_{m,l},m} + \eta)}{\Gamma(n^{(\cdot)}_{c_{m,l},-m} + n^{(\cdot)}_{c_{m,l},m} + W\eta)}\Big) \tag{2}$$

where $n^{(w)}_{c_{m,l},-m}$ is the number of instances of word $w$ assigned to the topic indexed by $c_{m,l}$ not including those in the current document.

Then we can get the posterior distribution for $c_m$:

$$\Pr(c_m|w, c_{-m}, z) \propto \Pr(w_m|c, w_{-m}, z)\Pr(c_m|c_{-m}) \tag{3}$$

where $\Pr(w_m|c, w_{-m}, z)$ is likelihood of the data given $c_m$ and a prior on $c_m$

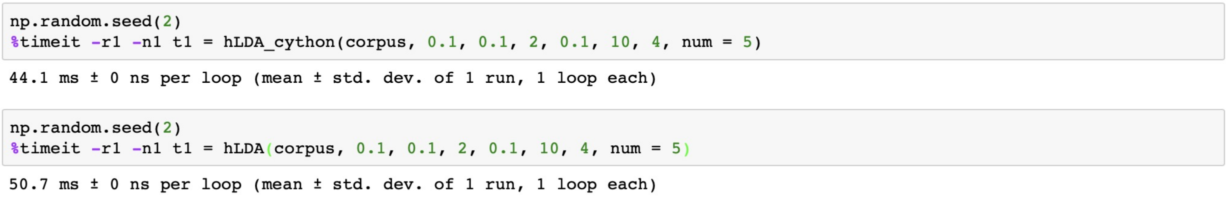
## 3 Optimization

We choose cython to optimize the code. First, we try to implement cython for all the functions. It turns out that cython works well speeding up single functions. However, when we rewrite the function, gibbs_sampling, the running time is similar or even slower since calling external cython function takes extra time. Therefore, we decide to only optimize functions gibbs_sampling and hLDA.

From the running time comparison, we can find out that the running time for hLDA_cython is 44.1 ms while it is 50.7 ms for the naive version. As a result, we speed up the process by $15\%$ using cython. The result is shown in Figure 3

## 4 A experiment on simulated data

In this section, we construct a simulated corpus by specifying the number of documents. We set the length of each document $d$ by randomly choosing a number from $n \in \{100, ..., 200\}$. We sample $w_d \sim N(0, 1)$ and use



```
np.random.seed(2)
%timeit -r1 -n1 t1 = hLDA_cython(corpus, 0.1, 0.1, 2, 0.1, 10, 4, num = 5)
```

```
44.1 ms ± 0 ns per loop (mean ± std. dev. of 1 run, 1 loop each)
```

```
np.random.seed(2)
%timeit -r1 -n1 t1 = hLDA(corpus, 0.1, 0.1, 2, 0.1, 10, 4, num = 5)
```

```
50.7 ms ± 0 ns per loop (mean ± std. dev. of 1 run, 1 loop each)
```

Figure 3: Comparison of hLDA and hLDA written by cython

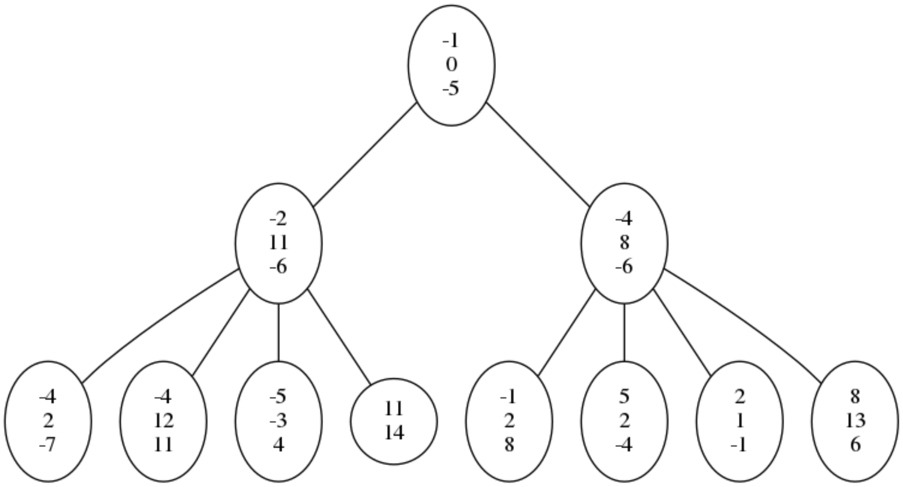


Figure 4: Tree plot of the topics generated from the simulation

'random.normal' function to get words. Here we use the number as the word. The normal distribution is symmetric bell shape that concentrates near mean 0. Therefore, most of the sampled words in the corpus should near 0. To make the result clearer, we multiply each sampled number by 10 and round to the nearest integer. For instance, we simulate 50 documents and apply hLDA. The output shows that hLDA returns values near mean 0, $\{-1, 0, 5\}$, in the root node and numbers such as $\{8, 13, 6\}$ far from mean in the leaves.

## 5 A experiment on real data

In this section, we apply hLDA to the real data. Since the real data used in the original paper is not accessible, we collect 150 paper abstracts from various fields such as chemistry and economics and get 10218 words. The data is saved as a text file in the 'code' folder. First We remove the stop words by using 'nltk' and then apply the hLDA model. With the pre-specified parameters, we get 3 levels with 21 nodes in total. For each node, we output the first six words as shown in Figure 5. The first node contains the words "dna", "repair", "growth", etc. We also visualize the result by a tree plot, thus we can find paths straightforwardly. In general, the output matches the abstracts topics and we can say that hLDA is helpful when dealing with text applications.

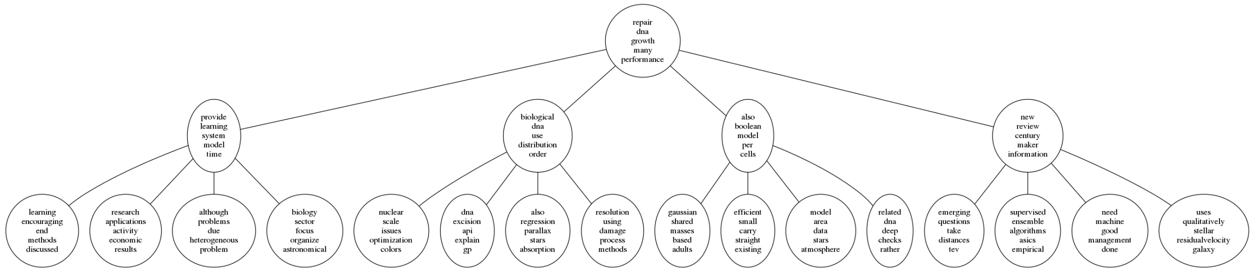

repair dna growth many performance
provide learning system model time
biological dna use distribution order
also boolean model per cells
new review century maker information
learning encouraging end methods discussed
research applications activity economic results
although problems due heterogeneous problem
biology sector focus organize astronomical
nuclear scale issues optimization colors
dna excision api explain gp
also regression parallax stars absorption
resolution using damage process methods
gaussian shared masses based adults
efficient small carry straight existing
model area data stars atmosphere
related dna deep checks rather
emerging questions take distances tev
supervised ensemble algorithms asics empirical
need machine good management done
uses qualitatively stellar residualvelocity galaxy

Figure 5: Tree plot of the topics generated from the real data

Table 1: Comparison of coherence

| Model | Coherence |
|---|---|
| hLDA | -15.36 |
| LDA | -10.37 |
| HDP | -22.31 |

## 6 Comparison analysis with competing algorithms

In this section, we compare three algorithms: LDA, hLDA, and HDP. hLDA and HDP are extensions of LDA, both of which take the advantage that the maximum number of topics can be unbounded and learned from the data rather than specified in advance. Compared with the hLDA which adopts a nonparametric prior via the nCRP, HDP doesn't describe the number of mixture components (topics) as a priori.

We import the 'LdaModel' and 'HdpModel' from 'gensim' and apply them to the real data that we described in the last section. The outputs are similar for the three algorithms, as "dna" and "repair" are shown in the first node of three outputs. However, comparing the running time, we can find that LDA and HDP generate the output faster than hLDA. We also incorporate the coherence to compare three models based on their interpretability. Here we only consider the intrinsic measure which compare a word only to the preceding and succeeding words respectively. The smaller the absolute value of coherence is, the better topics are distinguished. Table 1 lists the coherence of three models. LDA has the lowest coherence score and has the best performance. The coherence of hLDA model is not as good as LDA but much better than that of HDP.

In general, hLDA has a nice qualitative results for topic hierarchies and the inference of the number of topics is similar to HDP. However, hLDA restricts the document only follow a single path in the tree and it doesn't provide a strong quantitative evaluation.

## References

[1] D. Blei, T.Griffiths, M. Jordan, J. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. *Advances in neural information processing systems*, 2004.

[2] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993-1022, 2003.

[3] T. Griffiths and M. Steyvers. A probabilistic approach to semantic representation. *Proceedings of the 24th Annual Conference of the Cognitive Science Society*, 2002.