Summarizing Amazon Reviews using Hierarchical Clustering

Freddy Chong Tat Chua
Singapore Management University
freddy.chua.2009@smu.edu.sg

ABSTRACT

1. INTRODUCTION

Amazon is an internet website that sells a wide range of books from fictional novels to non-fictional scientific books. Amazon displays these books as individual web pages on their website. A typical web page of a book shows the title and brief product description of the book provided by the publisher. On the same web page, Amazon also allows any user to write their opinion and review of the book. These contributed reviews are shown on the same page for users to make a choice on whether to purchase the book. Since these reviews are subjective information, it is possible to score their relevance and quality through some means. Amazon provides such scoring base on a user voting system where users vote whether the review is helpful for making a purchase decision. The votes are then displayed with each review.

Such approach harness the collective intelligence of large number of users. Google PageRank also belongs to this category of relevance scoring. The number of users required for such an approach is huge because only a minority of users vote a review after reading it. The problem of not having sufficient users to vote a review is known as the cold start problem in recommendation systems.

1.1 Motivation

The cold start problem creates a vicious cycle of poor relevance score in recommendation systems. Consider the following example, suppose a particular book already has many reviews written for it. Many of these reviews have been voted and ranked according to their helpfulness votes. The reviews with many votes are ranked on top and reviews with no votes are ranked at the bottom. When a new review is written for this book, the new review which has no votes is ranked at the bottom. Since users typically do not spend time reading through all the reviews, new reviews are unlikely to be read by sufficient number of people which meant that it has low chances of obtaining any new votes. Such relevance scoring creates a vicious cycle where new reviews will never make their way to the top regardless of the review quality.

The commercial success of Google PageRank has greatly influenced the design of many information retrieval systems. These systems that harness the collective intelligence of users exhibit the major flaw which I described earlier. Ironically, it is the nature of human to favor the rich get richer phenomenon.

To solve such problems, we will take a step back and reexamine traditional approaches of textual analysis. Over the years, the introduction of statistics to the computer science community had provided computer scientists with a different approach of analyzing data. Statistics had already shown great usefulness in Information Retrieval. Google PageRank itself is a statistical process known as random walk. Term Frequency Inverse Document Frequency (TF-IDF) is also a statistical concept for weighting the importance of words in documents. More rigorous application of statistics are Topic Modeling such as Probabilistic Latent Semantic Analysis [4] and Latent Dirichlet Allocation (LDA) [1].

1.2 Methodology

I proposed to use an extended hierarchical version of LDA to analyze the text content of Amazon reviews [2]. The Hierarchical Latent Dirichlet Allocation (HLDA) extends the flat clustering algorithm of LDA to a tree-like hierarchical cluster. Clusters near the top of the tree represents topic of words which are general to all documents. These words are usually stop words. Clusters near the bottom of the tree are words of specialized topics. Ideally, this should be the word clustering we observe in HLDA.

There are several reasons for attempting to use HLDA.

1. The flat clustering of LDA do not reflect the importance of different clusters.

2. I hypothesize that reviews with a large proportion of words in the lower hierarchy offers a more unique viewpoint of the book content. Hence, we should favor reviews that concentrate most of the words in specialized clusters and exclude reading reviews that have a high proportion of words in the top clusters.

3. HLDA is a new statistical model which appear in the January 2010 issue of ACM Journal. Hence, I am curious about the performance and ability of this new model.

I had to implement my own HLDA algorithm for the experiments. I was initially reluctant to implement HLDA be-

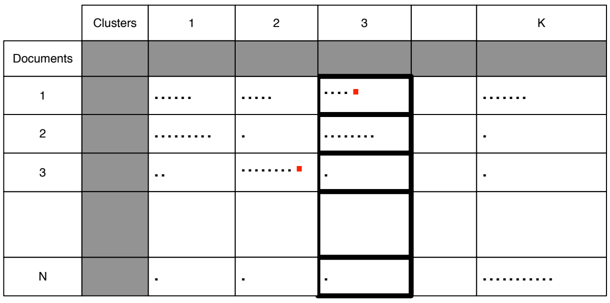

Figure 1: An Intuitive example of Text Clustering

cause I see it merely as a programming effort and reinvention of other people's work. I wanted to focus more on analysis of HLDA results rather than focusing on the engineering aspects of the algorithm. But after some investigation, I realised there are several important reasons for implementing the algorithm.

1. David Blei, the author of HLDA do not provide instructions on how to interpret the results of his implementation.
2. There is no known working implementation of HLDA since the paper recently appear in January 2010.
3. There is a text extraction toolbox called MALLET written by Andrew McCallum from University of Massachusetts (UMASS). The toolbox which have an implementation of HLDA is theoretically wrong. I realised it is wrong after spending days of code inspection.
4. Implementing the algorithm will allow me to understand the theoretical statistics at a fundamental level.
5. It is easier to engineer my own code for optimized performance of HLDA.

Due to constraints of computational resources, I restricted the set of Amazon reviews to only 236 reviews. I also modify Blei's HLDA such that it is suitable for a small corpus. The results obtained from the experiments show that we can indeed rank reviews using HLDA.

1.3 Organization

The paper is organized as follows. I will first briefly discuss the intuitive idea of LDA in Section 2.1. Then I will discuss the intuition of Infinite Gaussian Mixture Model which is crucial towards extending LDA to HLDA [5] in Section 3. Section 4 will illustrate HLDA. All of these discussion will focus on explaining the intuition and foregoing the rigor on the statistical formulation. Following that, Section 5 will discuss the experimental setup and results. I conclude the paper in Section 6.

2. LATENT DIRICHLET ALLOCATION

2.1 Fixed number of Clusters

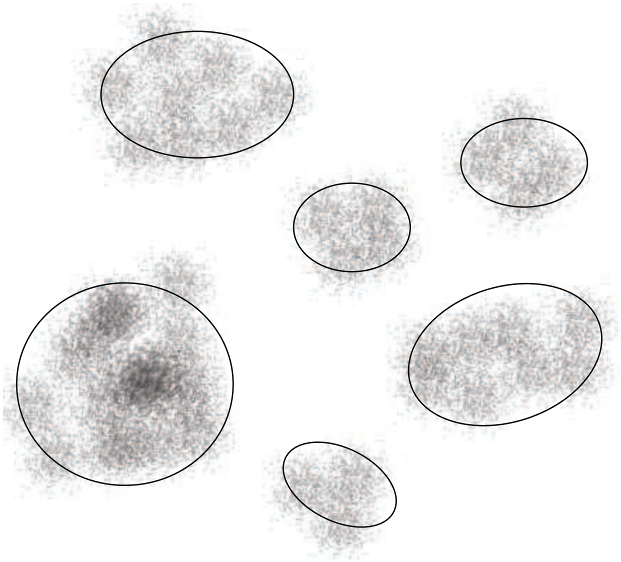

Figure 2: An example of Gaussian Mixture Model in Two Dimensional Space

LDA is a form of statistical clustering method that cluster words in documents. Each cluster of words is also known as topic. Refer to Figure 1 for an illustration of our following discussion. Suppose we have N documents and we divide the words in the corpus into K topics. Each row represents a document and each column represents a topic. The dots refer to the words of the documents and dots in each box represents the allocation of words to a certain topic. The words assigned in each topic addresses the synonymy problem. The red dots as shown are similar words that appear in different topics which reflects the polysemy problem.

2.2 Deciding the Number of Clusters

In most of the clustering algorithm that exists, the number of clusters has to be selected before executing the algorithm. Researchers who used LDA for their work had also pondered over the issue of how many topics to use. The problem of deciding the number of clusters is not a computational issue. Hence, computer science researchers have to find answers in the much older field of statistics. Indeed, in a related clustering problem, nonparametric statistics have been used to address the number of clusters issue for Gaussian distributed data. I give a brief summary of Infinite Gaussian Mixture Model here.

3. INFINITE GAUSSIAN MIXTURE MODEL

Infinite Gaussian Mixture Model is a generalization of Finite Gaussian Mixture Models. Refer to Figure 2 for an illustration of Mixture Models. Gaussian Mixture Model assumes that each point in the two dimensional space is drawn from a combination of K Gaussian Distribution, where each Gaussian Distribution represents a cluster. Before discussing Infinite Gaussian Mixture Model, I will first illustrate the Finite Gaussian Mixture Models here.

Suppose that in the two dimensional space, we have a set of N data points. We assume that there are K clusters in our data. Each data x_n is said to be drawn from one of the K clusters where each cluster is distributed according to a Gaussian Distribution with mean μ and covariance Σ.

$$x_n | k \sim \mathcal{N}(\mu_k, \Sigma_k)$$

The line reads as follows; given that the point x_n belongs to cluster k, then x_n is distributed with the mean μ_k and covariance Σ_k of cluster k. However, we do not know what are the cluster allocations for all of the N points. I shall briefly describe a Gibbs Sampling approach here.

3.1 Gibbs Sampling for Gaussian Mixture Model

1. For each x_n randomly assign a cluster k
2. For each x_n,
   (a) Unassign x_n from its current cluster
   (b) For each cluster k
      i. Compute its mean μ_k and covariance Σ_k base on current assignments.
      ii. Compute the probability $P(z_n = k)$ that x_n belongs to k.
   (c) Compute the cumulative probability $P(Z_n = k) = \sum_{j=1}^{k} P(z_n = k)$
   (d) Generate a random number $rand$
   (e) Assign cluster k to x_n if $rand \leq P(Z_n = k)$
3. Repeat Step 2 until the cluster assignments do not change much.

In Step 2.b.ii, $P(z_n = k)$ is calculated base on two factors,

1. The similarity of x_n with the mean and covariance of cluster k.
2. The popularity of cluster k.

To set the foundation for the infinite case of Gaussian Mixture Models, I will now illustrate the second factor, popularity of cluster k. The popularity of cluster k is computed as follows,

$$\frac{c_{n,k} + \frac{\alpha}{K}}{N - 1 + \alpha}$$

where $c_{n,k}$ is number of points allocated to cluster k. α is a smoothing parameter. The -1 is due to Step 2.a, the unassignment of x_n from its current cluster assignment. There is a mathematical derivation for the equation above, however, I will not go into details of that. In the following section, I will show that the Chinese Restaurant Process (CRP) extension of the popularity equation to infinite limit.

3.2 Chinese Restaurant Process

The previous equation assumes that there are only a fixed number of clusters K. In this section, we will introduce the Chinese Restaurant Process (CRP) for infinite clustering. The Chinese Restaurant Process is a special instance of Dirichlet Process which was rigorously proven in Ferguson 1973 paper [3].

Now imagine that K is a very large number. We will like to group all the empty clusters into a single cluster. Let $K*$ be the number of clusters that has allocated data. Then the number of clusters that has no allocation is given by $(K - K*)$. When we group all the empty clusters together, the popularity equation can be express in the following manner,

$$\frac{\frac{\alpha}{K}}{N - 1 + \alpha}(K - K*) = \frac{\alpha - \frac{\alpha K*}{K}}{N - 1 + \alpha}$$

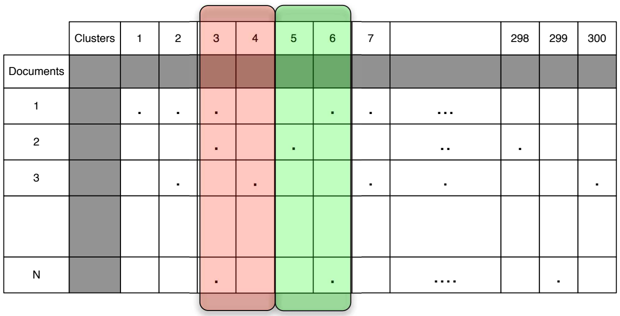

Figure 3: Infinite Limit of Text Clustering

When we let $K \lim \infty$, the above equation leads to

$$\frac{\alpha}{N - 1 + \alpha} \qquad (1)$$

The equation for clusters with allocations will then be

$$\frac{c_{n,k}}{N - 1 + \alpha} \qquad (2)$$

Hence, to extend the Gaussian Mixture Model to an infinite limit, we will modify step 2.b as follows

(b) For each cluster k
   ii. Compute the probability $P(z_n = k)$ that x_n belongs to k using Equation 2
   iii. Compute the probability $P(z_n = new)$ that x_n belongs to a new cluster using Equation 1

I have shown an algorithm of how we let the data decide the number of clusters. I tried to extend this simple method for determining the number of topics in text clustering using LDA. The next section will explain why this fails and Hierarchical Topic Modeling is the natural solution.

3.3 Extending LDA to Infinite Clusters

Figure 3 shows an example of what happens when I extend LDA to spawn new topics. The number of topics grow to a large number such that the words of the documents are no longer dominated by a few topics. The large number of topics lead to a sparse distribution of words among the topics. To correct this problem, Hierarchical Topic Modeling naturally becomes the solution. Several sparse topics can be merged to form single topics. These merged topics can be merged again to form larger topics.

4. HIERARCHICAL LATENT DIRICHLET ALLOCATION

Since Infinite Topic Modeling of Text does not work as simply as Infinite Gaussian Mixture Model, then using a hierarchical tree to group multiple topics is the natural solution. Here I present Blei's Hierarchical Latent Dirichlet Allocation (HLDA).

Refer to Figure 4. Although the motivation for hierarchical topic modeling was provided in previous section as a bottom up process, the HLDA as presented by Blei is a top down process. To perform hierarchical topic modeling we first need to have a tree. This tree is created by a stochastic process known as the Nested Chinese Restaurant

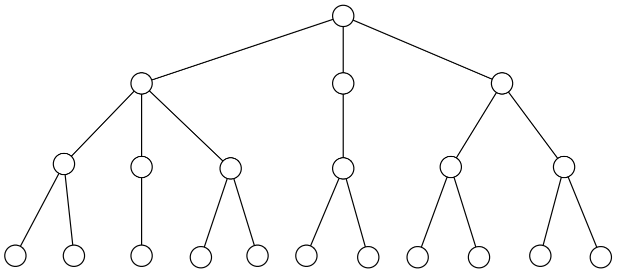

Figure 4: Tree of Hierarchical Clustering

Process (NCRP). Rigorous details of NCRP can be found in Yee Whye Teh's Hierarchical Dirichlet Process [6]. Recall in previous section that we went through the Chinese Restaurant Process (CRP) to stochastically spawn new clusters. NCRP is quite similar to CRP with the exception that each time a document is allocated to a topic, the document can spawn new topics within the previous topic. In other words, a document can spawn topic within topic. The following briefly illustrates the NCRP initialization,

1. We first start with a root node at the top of the tree.
2. For each document, let initial node be root,
   (a) Visit node.
   (b) Compute the probability $P(z_n = k)$ using Equation 2.
   (c) Compute the probability $P(z_n = new)$ using Equation 1.
   (d) Sample and decide whether to visit one of k existing child node or spawn new child node using above probabilities.
   (e) Once a node has been chosen, repeat steps (a) to (d) until the tree reaches a specified depth.
   (f) The nodes visited in the order as specified above constitutes a path in the tree.
   (g) For each word in a document, using the path chosen earlier,
      i. Randomly assign a node in the path to this word. In other words, the words are allocated to different levels in the tree.

Now that we have randomly created a tree, we shall proceed to perform the hierarchical topic modeling using gibbs sampling,

1. For each document,
   (a) Unassign document from its current path.
   (b) Choose a most likely path that this document should belong to.
   (c) Assign document to the chosen path.
   (d) For each word in a document,
      i. Unassign the word from its previous topic.
      ii. From the path chosen earlier, choose one of the nodes in this path that this word should belong to.
      iii. Words that are common to the entire corpus should be near the top, while words specific to this document should be near the bottom.
      iv. Assign the word to the chosen topic.
2. Repeat above for typically 1000 iterations.

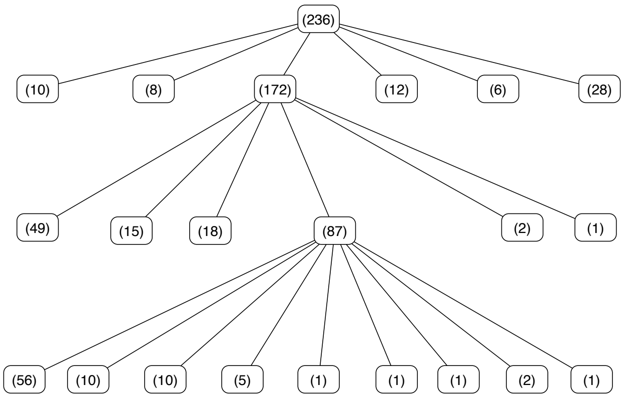

Figure 5: Blei's Hierarchical Tree

The HLDA as described above will automatically generate a tree of variable branch width but fixed height. Each node in the tree is not restricted to any specified number of child node. The NCRP and Sampling process will allow the algorithm to find the most suitable tree for the observed text corpus. I ran the experiments using this implementation of HLDA.

5. EXPERIMENTAL SETUP

Book ID	Book Title
0262032937	Introduction to Algorithms
0201000237	Data Structures and Algorithms
0201721481	C++ Primer
0137903952	Artificial Intelligence: A Modern Approach
020139829X	Modern Information Retrieval
0471117099	Applied Cryptography: Protocols, Algorithms, and Source Code in C
0072465638	Database Management Systems
0201385902	An Introduction to Database Systems
0321197844	An Introduction to Database Systems
0596100124	Database in Depth: Relational Theory for Practitioners
0201530821	Computational Complexity
0198538642	Neural Networks for Pattern Recognition

Table 1: Reviews of Books for Experiments

I use the reviews of Amazon books as shown in Table 1. The reviews were tested against two instance of HLDA. In the first instance, I use the original algorithm as described by Blei. In the second instance, I modify the algorithm such that the Hierarchical Tree is a fixed size constant binary tree.

5.1 Results of Original HLDA

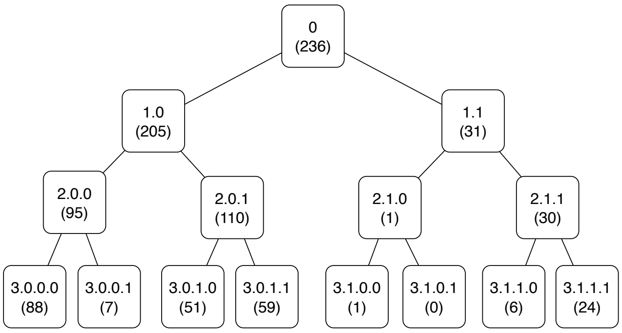

Figure 6: Hierarchical Binary Tree

Figure 5 shows the result of using Blei's algorithm to find the most suitable tree. I obtain a tree of size 78 nodes. Only part of the tree is shown here because the full tree has too many nodes to draw and display. The number in the nodes represent the number of documents which has the node in their path. Evaluation of this model is subjective and I invite readers to view the results which accompany this paper. The next section will show a modified implementation.

5.2 Results of Binary Tree HLDA

Figure 6 shows the result of the binary tree HLDA. The top number in each box represents the name of the node while the number in brackets show the number of documents which has the particular node in their path. There are a total of fifteen topics which is a reasonable number for such a small set of corpus. I attempt to rank the reviews according to the topic proportion. Initially, I had thought that reviews with a high proportion of words in the lower more specialized topics should be favored. However, upon observation of the results of HLDA, it turns out that reviews with high proportion of words in the top level topics should be favored. Since evaluation of Information Retrieval results has always been a tricky issue. The goodness of review cannot be measured by my own perception. Hence, the results of HLDA is included as files instead.

5.3 Discussion of Words in Topics

The top cluster appears to be a collection of stop words. The clusters below the top cluster has no obvious clustering of synonymous words. Perhaps HLDA is suitable for deciding what are the stop words in a corpus of documents. HLDA does not seem to obtain reasonable clustering of words for Amazon reviews because the reviews mainly consist of general words. Topic models or clustering is only useful when the corpus consist of documents which are specialized in discussing certain topics.

6. CONCLUSIONS

I have given an overall intuitive idea of recent advances in text clustering. The literature review summarizes the current state of the art techniques in statistical machine learning. For HLDA, I have shown that we can fix the tree structure and obtain reasonable results. Statistical machine learning has provided the research community with these models for analyzing our text corpus. While techniques such as dimension reduction or clustering exist, we still need to creatively apply such models to advance the artificial intelligence of our information systems. For LDA and HLDA, these models uses the bag-of-words assumption which loses most of the semantic meaning in sentences. More advance models such as Markov Models may be employed to preserve the temporal ordering of words in documents. Readers who are interested in knowing more about HLDA may refer to Blei's paper [2] and the last few pages shows my derivation of the equations which Blei had omitted.

7. REFERENCES

[1] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine learning Research*, 2003.

[2] D. M. Blei, T. L. Griffiths, and M. I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *J. ACM*, 57(2):1–30, 2010.

[3] T. S. Ferguson. A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973.

[4] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference*, pages 50–57. ACM Press, 1999.

[5] C. E. Rasmussen. The infinite gaussian mixture model. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *NIPS*, pages 554–560. The MIT Press, 1999.

[6] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

Derivation of Gibbs Sampling Equation for Hierarchical Latent Dirichlet Allocation

Freddy Chong Tat Chua
School of Information Systems
Singapore Management University
80 Stamford Road Singapore 178902
freddy.chua.2009@smu.edu.sg

1 Introduction

We let L denote the maximum depth of the tree. Each document takes a path in the tree where each path has a length of L. Suppose the tree has T number of nodes, there are potentially T possible paths for each document to choose from. There are two kinds of path, paths from root node to a leaf node and paths from root node to an internal node. When the internal node is chosen, we spawn new leaf nodes below.

1.1 Sampling the Path using Nested Chinese Restaurant Process

The nice property about tree is that there is only one path from root node to any other node in the tree. That means, there are T paths in the tree and we pick one out of T paths for the documents. The following shows the equation for performing the sampling,

$$P(c_d = t|w, c_{-d}, z, \eta, \gamma) \propto P(c_d|c_{-d}, \gamma)P(w_d|c, w_{-d}, z, \eta) \tag{1}$$

Two factors influence the probability that a document belongs to a path. The first factor is the number of documents allocated to a path, a document is more likely to belong to popular paths. The second factor is due to the likelihood of seeing the words in the document generated from a path. Equation 1 highlights these two factors.

1.1.1 Nested Chinese Restaurant Process

The $P(c_d|c_{-d}, \gamma)$ is a Nested Chinese Restaurant Process (NCRP). The NCRP is a special case of Dirichlet Process and its validity can be proven by the Kolmogorov Consistency Theorem. The Kolmogorov Consistency Theorem simply proves that clusters can be divided into sub-clusters. Let h_t denote the number of documents that had selected the path t. The NCRP can be described using the following,

$$P(c_d = t|c_{-d}, \gamma) = \frac{h_t}{N - 1 + \gamma} \tag{2}$$

$$P(c_d = new|c_{-d}, \gamma) = \frac{\gamma}{N - 1 + \gamma} \tag{3}$$

where N is the total number of documents in the corpus. Equation 2 shows the likelihood of choosing this node. Equation 3 is the likelihood of creating a new cluster at this level.

Suppose the sampled node is an internal node instead of a leaf node, then it means we spawn new leaf nodes until we reach the maximum depth as defined. Suppose for a given path c_d, the path has topic levels $1, \ldots, K$ and there are T number of word topic distributions. Let's use the following,

$$P(w_{d,n} = v|c_d, z_{d,n}, B) = b_{t,k,v} \tag{4}$$

It is important to note here that t denote the path and k selects the level in the path, a tuple (t, k) selects a topic.

$$B_{t,k} = (b_{t,k,1}, \ldots, b_{t,k,V}) \tag{5}$$

$$P(B_{t,k}|\eta) = \frac{\Gamma(\sum_{v=1}^{V} \eta)}{\prod_{v=1}^{V} \Gamma(\eta)} \prod_{v=1}^{V} b_{t,k,v}^{\eta-1} \tag{6}$$

Now that the basic distributions and definitions are there, we shall proceed to do some tough stuff.

$$\begin{aligned}
P(w_d|c_d, z_d, \eta) &= \int P(w_d, B|c_d, z_d, \eta)\, dB && (7)\\
&= \int P(w_d|c_d, z_d, B) P(B|\eta)\, dB && (8)\\
&= \int \left[\prod_{n=1}^{N} P(w_{d,n}|c_d, z_{d,n}, B) \right] P(B|\eta)\, dB && (9)\\
&= \int \left(\prod_{t=1}^{T} \prod_{v=1}^{V} b_{t,k,v}^{f_{t,k,v}} \right) P(B|\eta)\, dB && (10)\\
&= \prod_{t=1}^{T} \left[\frac{\Gamma(V\eta)}{\Gamma(\eta)^V} \frac{\prod_{v=1}^{V} \Gamma(f_{t,k,v} + \eta)}{\Gamma(V\eta + \sum_{v=1}^{V} f_{t,k,v})} \right] && (11)\\
&\propto \prod_{t=1}^{T} \frac{\prod_{v=1}^{V} \Gamma(f_{t,k,v} + \eta)}{\Gamma(V\eta + \sum_{v=1}^{V} f_{t,k,v})} && (12)
\end{aligned}$$

Now for even tougher stuff,

$$\begin{aligned}
P(w_d|c_d, w_{-d}, z_d, \eta) &= \frac{P(w|c_d, z_d, \eta)}{P(w_{-d}|c_d, z_d, \eta)} && (13)\\
&= \prod_{t=1}^{T} \left[\frac{\Gamma\left(V\eta + \sum_{v=1}^{V} g_{t,k,v}\right)}{\prod_{v=1}^{V} \Gamma(\eta + g_{t,k,v})} \frac{\prod_{v=1}^{V} \Gamma\left(g_{t,k,v} + f_{t,k,v} + \eta\right)}{\Gamma\left(V\eta + \sum_{v=1}^{V} \left(g_{t,k,v} + f_{t,k,v}\right)\right)} \right] && (14)
\end{aligned}$$

And expressing in Logarithm form,

$$\log P(w_d|c_d, w_{-d}, z_d, \eta) = \log \left[\prod_{t=1}^{T} \left[\frac{\Gamma\left(V\eta + \sum_{v=1}^{V} g_{t,k,v}\right)}{\prod_{v=1}^{V} \Gamma(\eta + g_{t,k,v})} \frac{\prod_{v=1}^{V} \Gamma\left(g_{t,k,v} + f_{t,k,v} + \eta\right)}{\Gamma\left(V\eta + \sum_{v=1}^{V} \left(g_{t,k,v} + f_{t,k,v}\right)\right)} \right] \right] \tag{15}$$

$$\begin{aligned}
&= \sum_{t=1}^{T} \left[\log \Gamma \left(V\eta + \sum_{v=1}^{V} g_{t,k,v} \right) - \sum_{v=1}^{V} \log \Gamma(\eta + g_{t,k,v}) \right.\\
&\qquad \left. + \sum_{v=1}^{V} \log \Gamma\left(g_{t,k,v} + f_{t,k,v} + \eta\right) - \log \Gamma \left(V\eta + \sum_{v=1}^{V} \left(g_{t,k,v} + f_{t,k,v}\right) \right) \right]
\end{aligned} \tag{16}$$

When sampling whether to branch off, the equations look like the following, it is pretty similar to the one above except that $g_{t,k,v}$ is always zero.

$$\log P(w_d|c_d, w_{-d}, z_d, \eta) = \log \left[\prod_{t=1}^{T} \left[\frac{\Gamma\left(V\eta + \sum_{v=1}^{V}\right)}{\prod_{v=1}^{V} \Gamma(\eta)} \frac{\prod_{v=1}^{V} \Gamma\left(f_{t,k,v} + \eta\right)}{\Gamma\left(V\eta + \sum_{v=1}^{V} f_{t,k,v}\right)} \right] \right] \tag{17}$$

$$\begin{aligned}
&= \sum_{t=1}^{T} \left[\log \Gamma\left(V\eta\right) - \sum_{v=1}^{V} \log \Gamma(\eta) \right.\\
&\qquad \left. + \sum_{v=1}^{V} \log \Gamma\left(f_{t,k,v} + \eta\right) - \log \Gamma \left(V\eta + \sum_{v=1}^{V} f_{t,k,v} \right) \right]
\end{aligned} \tag{18}$$

1.2 Sampling the Topics in the Path using Stick Breaking Construction

Suppose we have D documents and (d, N) words in document d. For each word n in document d, we choose the topic it belongs to as follows,

$$V_i \sim Beta(m\pi, (1-m)\pi) \tag{19}$$

$$P(z_{d,n} = k|V_1, \ldots, V_k) = V_k \prod_{i=1}^{k-1}(1-V_i) \tag{20}$$

Suppose we let $e_{d,k}$ denote the counts of occurrence for each k in document d. Then suppose we want to derive the posterior distribution of $V_1, \ldots, V_k$

$$V_1|e_{d,1}, \ldots, e_{d,K} \sim Beta\left(m\pi + e_{d,1}, (1-m)\pi + \sum_{i=2}^{K} e_{d,i}\right) \tag{21}$$

$$V_2|e_{d,1}, \ldots, e_{d,K} \sim Beta\left(m\pi + e_{d,2}, (1-m)\pi + \sum_{i=3}^{K} e_{d,i}\right) \tag{22}$$

$$V_k|e_{d,1}, \ldots, e_{d,K} \sim Beta\left(m\pi + e_{d,k}, (1-m)\pi + \sum_{i=k+1}^{K} e_{d,i}\right) \tag{23}$$

Hence,

$$\begin{aligned} P(z_{d,n} = k|z_{d,-n}, m, \pi) &= E\left[V_k \prod_{i=1}^{k-1}(1-V_i)\right] && (24)\\ &= \frac{m\pi + e_{d,k}}{\pi + \sum_{i=k}^{K} e_{d,i}} \prod_{i=1}^{k-1} \frac{(1-m)\pi + \sum_{j=i+1}^{K} e_{d,j}}{\pi + \sum_{j=i}^{K} e_{d,j}} && (25)\end{aligned}$$

As for the word, it goes as follows, suppose we have V number of words in the vocabulary, let $d_{k,v}$ be the number of times word v is allocated to topic k.

$$P(w_{d,n} = v|z, c, w_{d,-n}) = \frac{d_{k,v} + \eta}{\sum_{v=1}^{V} d_{k,v} + V\eta} \tag{26}$$

So to sample a topic, the expression is as follows

$$P(z_{d,n} = k|z_{d,-n}, c, w, m, \pi, \eta) = \left[\frac{m\pi + e_{d,k}}{\pi + \sum_{i=k}^{K} e_{d,i}} \prod_{i=1}^{k-1} \frac{(1-m)\pi + \sum_{j=i+1}^{K} e_{d,j}}{\pi + \sum_{j=i}^{K} e_{d,j}}\right] \frac{d_{k,v} + \eta}{\sum_{v=1}^{V} d_{k,v} + V\eta} \tag{27}$$