1 Flat Topic Model

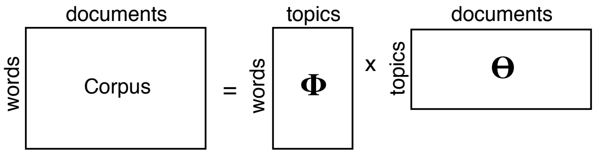

Figure 1: Topic Model

Latent Dirichlet Allocation LDA topic analysis uses a per document bag of words approach to determine topic compositions of words and document mixtures of topics. Figure 1 from **Steyvers:2007** explains a corpus as the product of topic compositions (Φ) and document mixtures (Θ). Compositions are interpreted as topics or themes across documents, conversations, or discussions. Document mixtures can be examined to see how mixture proportions vary by document or even over time. Topic analysis reduces the dimensionality of a corpus by orders of magnitude from millions or billions of words to frequency distributions of tens, hundreds, or thousands of topics.

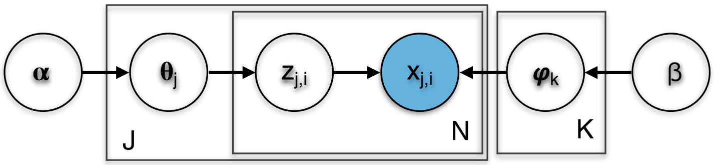

Figure 2: LDA Plate Model

LDA (**Blei:2003**) is based on a Bayesian generative probabilistic model portrayed as a plate graph in figure 2. Mixtures, θ_j, for J documents, and compositions, ϕ_k, for K topics, are generated by Dirichlet distributions with parameters α and β respectively. Topic assignments are sampled from the mixture for document j categorical, $z_{j,i} \sim \text{Categorical}(\theta_j)$, and term assignments are sampled from the topic composition for topic $z_{j,i}$ categorical, $x_{j,i} \sim \text{Categorical}(\phi_{z_{j,i}})$, for each of the N_j document word positions.

The joint distribution of words and topics is given by $P(\boldsymbol{x}, \boldsymbol{z}) = P(\boldsymbol{x}|\boldsymbol{z})P(\boldsymbol{z})$ with

$$P(\boldsymbol{x}|\boldsymbol{z}) = \left(\frac{\Gamma\left(V\beta\right)}{\Gamma\left(\beta\right)^V}\right)^K \prod_{k=1}^{K} \frac{\prod_v \Gamma\left(n_{\bullet kv} + \beta\right)}{\Gamma\left(n_{\bullet k \bullet} + V\beta\right)} \quad (1) \quad P(\boldsymbol{z}) = \left(\frac{\Gamma\left(\alpha_\bullet\right)}{\prod_k \Gamma\left(\alpha_j\right)}\right)^J \prod_{j=1}^{J} \frac{\prod_k \Gamma\left(n_{jk\bullet} + \alpha_k\right)}{\Gamma\left(n_{j\bullet\bullet} + \alpha_\bullet\right)}, \quad (2)$$

where K is the number of topics, V is the vocabulary size, $n_{\bullet kv}$ is the number of times word v and topic k occur together, $n_{\bullet k\bullet}$ is the frequency of topic k, β is the symmetric vocabulary prior, and $\Gamma\left(\,\right)$ is the gamma function; J is the number of documents, $n_{jk\bullet}$ is the number of times topic k and document j

occur together, $n_{j\bullet\bullet}$ is the frequency of document j, and α_k is the asymmetric topic prior.

Analysis reverses the generative model inferring latent topics $z_{j,i}$, document topic mixtures θ_j, and topic word compositions ϕ_k from the document words $x_{j,i}$, and α and β probability priors. Topic analysis identifies topic compositions and document mixtures from a corpus. Gibbs sampling (**Gelman:2004**) assigns topics to text in the training corpus considering each text word in turn. The Gibbs estimator based on (**Griffiths:2004**) for individual topics is:

$$P(z_{j,i} = k|\boldsymbol{z}^{-i}, \boldsymbol{x}) \propto \frac{n^{-i}_{\bullet k x_{ji}} + \beta}{n^{-i}_{\bullet k \bullet} + V\beta} \frac{n^{-i}_{jk\bullet} + \alpha_k}{n^{-i}_{j\bullet\bullet} + \alpha_\bullet}, \tag{3}$$

where the $^{-i}$ indicates that the current term and topic assignment are excluded.

Estimators of ϕ and θ, when needed, look like factors of the Gibbs sampler using word type v instead of sampled word $x_{j,i}$ and without the Gibbs sampling exclusion.

$$\hat{\phi}_{vk} = \frac{n_{\bullet kv} + \beta}{n_{\bullet k\bullet} + V\beta} \tag{4}$$

$$\hat{\theta}_{kj} = \frac{n_{jk\bullet} + \alpha_k}{n_{j\bullet\bullet} + \alpha_\bullet} \tag{5}$$

We monitor training progress with $\log P(\boldsymbol{x}, \boldsymbol{z})$. Typically, after sufficient iterations, $\log P(\boldsymbol{x}, \boldsymbol{z})$ converges to steady state and the simulation terminates. Analysis products are topic determinations for the corpus as well as estimates of topic word compositions Φ and document topic mixtures Θ.

We measure reproducibility with test log likelihood. Topics Φ from training are used to infer document mixtures Θ on the test corpus, and log $P(\boldsymbol{x})$, test log likelihood, is calculated from **TehYee:2007a** as:

$$P(\boldsymbol{x}^{test}) = \prod_{j,i} \left(\sum_k \frac{n_{\bullet k x_{ji}} + \beta}{n_{\bullet k \bullet} + V\beta} \frac{n^{test}_{jk\bullet} + \alpha_k}{n^{test}_{j\bullet\bullet} + \alpha_\bullet} \right), \tag{6}$$

where the sum is over all topics per word and the product is over all words and documents. Perplexity, $Perplexity(\boldsymbol{x}) = \mathrm{e}^{(-logP(\boldsymbol{x}))/N}$, indicates the uncertainty of predicting individual words, where the maximum perplexity corresponds to the vocabulary size.

2 Hierarchical Topic Model

Generate Topic Tree We generate the topic tree modeling the corpus topic tree structure. This topic tree structure includes the hierarchical relation between topics, a Dirichlet process (DP) at each tree node for selecting individual topics, topic compositions for words (terms) for each tree node, and the tree root. While the corpus tree is generated in the abstract here, during inference the tree structure, corpus topic weights, topic compositions are inferred from the corpus.

Construct a global topic tree $\mathcal{T}$ rooted at node h_0. Sample topic ϕ^{**} drawn from H, $\phi^{**} \sim H$, and assign it to the root node h_0. Sample a Dirichlet process G, itself drawn from a DP, $G|\gamma, H \sim DP(\gamma, H)$, with the same continuous base distribution H, and assign it to the root node h_0. Even though H is continuous, the derived Dirichlet process (DP) is discrete with probability one. The Dirichlet process G generates topic samples $\phi \sim G|\gamma, H$. Collapsing the samples of ϕ to unique discrete values yields ϕ^{**} the *atoms* of G. These *atoms* become topics of the child nodes with probability given by the weight of the collapsed nodes of G.

Recursively sample the tree nodes starting with the root node h_0. Define child nodes each consisting of a unique topic sampled from the parent Dirichlet process, $\phi^{**} \sim G_h|\gamma, H$, and a new Dirichlet process (DP) sampled for this node, $G|\gamma, H \sim DP(\gamma, H)$. A theoretical infinitely branching and deep topic tree is constructed in this manner.

Designate the path to a node at level l by $\mathbf{h}_l = (h_0, \ldots, h_l)$ where the initial node for each path is the root h_0. Designate specific nodes, topics, and Dirichlet processes by $\mathbf{h}_l$, $\phi^{**}_{\mathbf{h}_l}$, and $G_{\mathbf{h}_l}$. Since the base distribution H is continuous, the root topic $\phi^{**}_{\mathbf{h}_0}$ and the unique topics $\phi^{**}_{\mathbf{h}_l}$ for each Dirichlet process are unique over the entire topic tree $\mathcal{T}$.

For topic analysis of text, I define the base distribution as a symmetric Dirichlet distribution with concentration η and dimension V (the vocabulary size), $H \sim Dirichlet(\eta/V, \mathbf{1}_V)$. This is what was used for LDA topic analysis and is consistent with **Paisley:2015**

The global tree generative algorithm 1 constructs the tree structured hierarchy of topics and DPs. The root node h_0 defines a discrete probability distribution of the topic word composition ϕ^{**} and a Dirichlet process, G, that links to next level nodes. From the root node, the algorithm recursively constructs sub-trees of the global tree $\mathcal{T}$.

Generate Document Tree and Document Given the corpus topic tree we generate the document tree and the document composition of topics and words for each document. Document tree structure is shared with the corpus tree as are the topic compositions. Dirichelet processes (DPs) selecting topics from the tree are derived from corresponding Dirichlet Processes in the corpus tree. While topic weights

Algorithm 1 Generate Global Tree

```
**GenerateGlobalTree**
  $H \leftarrow \text{Dirichlet}(\eta/V, \mathbf{1}_V)$
  $\phi^{**} \leftarrow$ draw $H$
  $G \leftarrow$ draw $\text{DP}(\gamma, H)$
  $h_0 \leftarrow (\phi^{**}, G)$
  ConstructGlobalTree($h_0$)
  $\mathcal{T} \leftarrow h_0$
**ConstructGlobalTree**(TreeNode $h$)
  **for** unique $\phi^{**}$ from $G_h$ **do**
    $G \leftarrow$ draw $\text{DP}(\gamma, H)$
    $h' \leftarrow (\phi^{**}, G)$
    ConstructGlobalTree($h'$)
    add child $h'$ to parent $h$
  **end for**{Infinitely branching and deep generative process}
```

of DPs, topics, and terms are generated in the abstract here, during inference §**??** they are inferred from individual documents and the corpus as a whole.

Given the global tree of unique topics ϕ^{**} and Dirichlet processes G sampled from the base distribution H, generate individual documents. For each document, generate a hierarchical document topic tree, and then generate individual topic assignments and words. Document tree generation is based on the global topic tree, replacing the Dirichlet process of each node with a corresponding document Dirichlet process sampled from the global Dirichlet process, $G_{j,\mathbf{h}_l} \sim DP(\alpha, G_{\mathbf{h}_l})$. The *atoms* of the document Dirichlet processes come from the corresponding global Dirichlet processes and so are the same, but the probabilities are a re-weighting of the *atoms* of the corresponding $G_{\mathbf{h}_l}$ (**TehYee:2010b**).

For each position (planned word) of a document sample the topic ϕ^{**} from the document tree h_0. The word is sampled as multinomial from the topic ϕ^{**} of the sampled node.

The document generative algorithm 2 makes explicit this definition of a document tree structured hierarchy of Dirichlet processes, and the generation of topics and words for a document. For each document j generate the corresponding document tree of topic nodes $\mathcal{T}_j$. Each node of the document tree defines a Dirichlet process (DP) sampled from the DP of the corresponding node of the global tree, and references a topic composition ϕ^{**} from the corresponding node of the global tree. Generate the topics and words: For each position in the document sample the topic instance $\phi_{j,i}$ from the document topic tree and then sample the word instance $x_{j,i}$ from the topic composition.

Algorithm 2 Generate Document

```
GenerateDocument $j$
  $G \leftarrow$ draw DP($\alpha, G_{h_0}$)
  $h_{j,0} \leftarrow (\phi_{h_0}, G)$
  ConstructDocumentTree($h_0, h_{j,0}$)
  $\mathcal{T}_j \leftarrow h_{j,0}$
  for each position i do
    $\phi_i \leftarrow$ DrawTopic($h_{j,0}$)
    $x_i \leftarrow$ draw Categorical($\phi_i$)
  end for
ConstructDocumentTree(TreeNode $h$, DocTreeNode $h_j$)
  for children $h'$ of $h$ do
    $G \leftarrow$ draw DP($\alpha, G_{h'}$)
    $h'_j \leftarrow (\phi^{**}_{h'}, G)$
    $h_j$ add child $h'_j$
    ConstructDocumentTree($h', h'_j$)
  end for{Generative process runs in infinite time}
DrawTopic(DocTreeNode $h_j$)
  $\theta \leftarrow$ flatten document tree DP hierarchy
  $\phi^{**} \leftarrow$ draw Categorical($\theta$)
  return $\phi^{**}$
```

Inference on Hierarchical Model Let us turn the generative model inside-out. Instead of generating a corpus, we determine the corpus topic tree, $\mathcal{T}$, with its hierarchical structure and topic word compositions. Instead of generating documents, we determine the document topic trees, $\mathcal{T}_j$, with their topic mixtures, and latent topics corresponding to word positions, $\phi_{j,i}$.

The path notation, $\mathbf{h}_l$, shows the parent child relationship between topic nodes, but it is unwieldy. So I index the topic nodes over the global tree, $k = 0, \ldots, \infty$, and let the index stand in for the full path name.

When the global and document topic trees are inferred from a text corpus by Gibbs sampling, the model is restricted to a maximum level, L, maximum topic node index, K, and finite amount of text, N, thus **not** requiring infinite time to reverse engineer the topic tree. Key aspects of Gibbs sampling used here are (**Gelman:2004**; **TehYee:2010b**): (1) Draw the latent topic for the current term after excluding the current topic and term from frequency counts; (2) Incorporate probability parameters from the generative model into estimators as priors; and (3) Simulate via collapsed Gibbs sampling the estimating of topic compositions and document topic mixtures on demand.

Topic Composition Estimator The estimator for the topic composition is similar to LDA (**Griffiths:2004**)

$$Pr(x_{j,i} = v | z_{j,i} = k) = \begin{cases} \dfrac{n^{-i}_{\bullet,k,v} + \eta}{n^{-i}_{\bullet,k,\bullet} + V\eta} & \text{for } k \text{ already defined} \\ 1/V & \text{for } k = k^{new} \end{cases}$$

except that with hierarchical topic analysis there is always the possibility to sample an entirely new topic, k^{new}.

Topic Mixture Estimator **TehYee:2006** gives the Gibbs sampler for the hierarchical Dirichlet process (DP). I've adapted the estimator to hierarchical topic analysis where divisors for the estimators are sums over counts of children of the current node. This difference is critical because in the hierarchical topic model, there is a DP for each document tree node and the count for any particular DP is not fixed as it is in the simpler HDP.

$$Pr(z_{j,i} = k | k', \mathbf{z}_j^{-i}, \mathbf{m}, \alpha, \mathcal{T}_j, \gamma, \mathcal{T}) = \begin{cases} \dfrac{n^{-i}_{j,k,\bullet} + \alpha \left({}^{m_k^{-i}}/_{(\gamma + M'_k)} \right)}{\alpha + N'_k} & \text{for } k \in \mathcal{T} \\ \dfrac{\alpha \left({}^{\gamma}/_{(\gamma + M'_k)} \right)}{\alpha + N'_k} & \text{for } k^{new} \end{cases} \tag{7}$$

where k' is the current node, k is an index referring either to an exiting topic in $\mathcal{T}$ that is a child of k' or a new topic, $\mathbf{z}_j$ is the topic vector for document j, $\mathbf{m}$ is the vector of topic 'table' counts, α is the weight for sampling from corpus $\mathcal{T}$, γ is the weight for sampling a new corpus topic, and $\mathcal{T}_j$ is the topic tree for document j. M'_k and N'_k are defined as follows

$$M'_k = \sum_{r \in child(k')} m_r^{-i} \quad \text{and} \quad N'_k = \sum_{r \in child(k')} n^{-i}_{j,r,\bullet}$$

where the function $child(k')$ returns all the children of k'.

The use of 'table' comes from the Chinese restaurant metaphor and captures the event where a client sits at a new table in the restaurant, and orders a dish already being served in the restaurant chain and maybe even in the same restaurant. We use (**TehYee:2006**)'s direct assignment method of determining $\mathbf{m}$. For that matter, the factors ${}^{m_k^{-i}}/_{(\gamma + M'_k)}$ and ${}^{\gamma}/_{(\gamma + M'_k)}$ are replaced by the vector β estimated as in (**TehYee:2006**).

For each topic decision, the current node starts at the global topic tree root and considers all nodes accessible from the document tree, i.e. with $n_{j,k'} > 0$ or a child of node with $n_{j,k'} > 0$. The probability of traversing a path to topic k from the root is the product of the node probabilities along the path:

$$\rho_k = \prod_{k'=path(k)} Pr(k'|prev(k'), \mathbf{z}_j^{-i}, \mathbf{m}, \alpha, \mathcal{T}_j, \gamma, \mathcal{T}),$$

where the root node is in *all* paths with probability one and $prev(k')$ is initially set to the root.

If the topic is present in the document tree ($n_{j,k} > 0$ excluding the current $z_{j,i}$), then both the topic count for the document tree and the α weighted proportion of all documents that have this topic contribute to the probability of selecting the topic k. If only the global tree includes this topic, then only the α weighted proportion of all documents with this topic contributes to the probability of selecting this topic. Entirely new topics are created as a child of node $prev(k')$ with a weight proportional to $\alpha\gamma$. If the model were simply that of a HDP, we could dispense with the divisor $\alpha + N'_k$, but as topics come from various branches of the topic tree, the divisor varies as well.

Combined Gibbs Sampler We combine the topic composition and mixture estimators to form the Gibbs sampler for choosing a new topic:

$$Pr(k|x_{j,i} = v) \propto Pr(x_{j,i} = v|z_{j,i} = k) \cdot \rho_k.$$

Practical considerations in development of the Gibb's sampler for hierarchical topic analysis are

1. Topics are added to the document tree only in the neighborhood of the existing document nodes.
2. A leaf document node is deleted when it is no longer being used.
3. Corpus topics are deleted when no longer used.

Test Log Likelihood To assess how good the topic model is (or whatever model) it is important to measure how such model fits with a corpus other than that used for training, because assessing model fit on the same corpus on which is was trained (optimized) gives a biased result. So a test corpus is used to assess the goodness of the model and to make comparisons between models trained on similar corpora.

I follow (**TehYee:2007a**; **Paisley:2015**) in computing log likelihood on the test corpus where test and training documents were randomly split from a general corpus resulting in separate training and test corpora. This approach assures that training and test corpora are independent, but that there are likely no other substantial differences between the corpora. Predictive log likelihoods on test (held-out) corpora are computed based on the trained topic model. Test log likelihood results can be compared for different analysis parameters (e.g. priors), repeat runs, or a different methodology (e.g. LDA topic analysis). The detail process for determining predictive log likelihood on test corpora follows:

1. Split the corpus x ways into training and test corpora.
2. Analyze the x training corpora reserving the topic compositions ϕ^{**}, corpus topic structure $\mathcal{T}$, and corpus document topic counts $\mathbf{m}$.
3. Analyze the x test corpora holding constant the reserved structures from training using $p\%$ of each test document in the analysis to estimate document parameters.
4. Use the remaining $(100-p)\%$ of the observations per test document to compute the test log likelihood.
5. The test log likelihood calculation can be done in parallel with analysis of the test documents, so multiple S steady state measurements can be gathered for each test document. Using multiple measurements reduces the measurement error of the test log likelihood.

Flat Test Log Likelihood This formula for estimating the test corpus probability for *flat* models is adapted from **TehYee:2007a**

$$Pr(\mathbf{x}^{test}) = \prod_{j,i} \frac{1}{S} \sum_{s=1}^{S} \sum_{k} \theta^{s}_{j,k} \phi_{k,x^{test}_{j,i}} \quad \text{where } \phi_{k,v} = \frac{n^{train}_{\bullet,k,v} + \eta}{n^{train}_{\bullet,k,\bullet} + V\eta} \quad \text{and } \theta^{s}_{j,k} = \frac{n^{test,s}_{j,k} + \alpha(\frac{m_k}{m_\bullet})}{n^{test}_{j,\bullet} + \alpha} \tag{8}$$

The topic compositions, ϕ, are determined from the training corpus, and so do not vary with the test corpus. The document topic mixtures, θ, are determined for the test corpus. Because calculating $Pr(\mathbf{x}^{test})$ will underflow for reasonable sized test sets, log likelihood (LL) is calculated instead

$$LL(\mathbf{x}^{test}) = \sum_{j,i} \ln\Big(\frac{1}{S} \sum_{s=1}^{S} \sum_{k} \theta^{s}_{j,k} \phi_{k,x^{test}_{j,i}}\Big), \tag{9}$$

with each document calculated separately and the log likelihoods summed over documents.

The formulas above estimate the likelihood of the test corpus given the collection of topics. It pays no attention to the hierarchical structure of topics.

Hierarchical Test Log Likelihood Predictive log likelihood, test LL(x), in the hierarchical case follows **TehYee:2007a** except that mixture weights, $\theta^{s}_{j,k}$, are no longer from a *flat* model, and the resulting log likelihood must be standardized for the number of hierarchy levels, $L+1$. We estimate the weight of sampling document topic *j,k* hierarchically as

$$\theta^s_{j,k} = \frac{\alpha\pi^s_k + n^s_{j,k}}{\alpha + n^s_j}\theta^s_{j,par(k)}$$

where π_k represents the DP corpus probability, par(k) identifies the parent DP of topic *k*, the root probability is one, and *s* indexes a test sample interval. The weights $\theta^s_{j,par(k)}$ haven't been normalized and they sum to L+1, the number of hierarchy levels. Normalization results in a constant correction to the test LL calculation

$$LL(\mathbf{x}^{test}) = \sum_{j,i} \ln(\frac{1}{S}\sum_{s=1}^{S}\sum_{k}\theta^s_{j,k}\phi_{k,x^{test}_{j,i}}) - ln(L+1) \tag{10}$$

3 Hierarchical Topic Model Theory

The term *hierarchical* is used both in the sense of probability distributions, where one distribution is derived from a parent distribution, and in the sense of topic structures, where global and document topic structures appear as topic hierarchies (i.e. topic trees). I start this review of theory with the Dirichlet process (DP) model. This discussion borrows from **TehYee:2006**; **TehYee:2010b**

Dirichlet Process Model The DP model handles the situation where compositions of basic components are allocated to individual instances in proportion to potentially infinite list of mixture weights. Examples are a picture as composition of many different image components, or a restaurant serving clients from a menu, fertilizers made to certain formulas, or words from a document chosen from various topics; where in each case the number of image component types, menu options, fertilizer products, or topics is not specified in advance.

We observe a collection of values $\{x_1, \ldots, x_n\}$ with corresponding latent parameters $\{\phi_1, \ldots, \phi_n\}$, where "each ϕ_i is drawn independently and identically (iid) from G, while each x_i is distributed as $F(\phi_i)$ parameterized by ϕ_i" (**TehYee:2010a**). This is summarized as

$$x_i|\phi_i \sim F(\phi_i),\ \phi_i|G \sim G,\ G|\gamma, H \sim DP(\gamma, H).$$

H is a continuous distribution, G is a DP and is discrete with probability one, and so multiple draws of ϕ_i can take on the same discrete value. We can think of the ϕ_i as designating a cluster's parameters, and of the x_i as the observed values of the cluster. Specifically as used here for topic analysis, the ϕ_i correspond to multinomial probabilities drawn from a DP for selecting words x_i from the topic parameterized by ϕ_i, where the DP is formed from the concentration parameter γ and base distribution H. Here we take H as the symmetric Dirichlet distribution with concentration parameter η and number of categories W (e.g., the vocabulary size) and $\phi_k^{**} \sim Dirichlet(\eta/W, \mathbf{1}_W)$.

The Dirichlet process G can also be written as:

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k^{**}},$$

where ϕ_k^{**} signifies the unique values sampled from H as $\phi_k^{**}|H \sim H$ and ϕ_i were sampled from values of ϕ^{**} with probability π_k where π_k is sampled from the *stick breaking* construction with parameter γ.

The stick breaking construction formalizes how the Dirichlet process can be constructed. The probability π_k of selecting a particular component indexed by k is the product of probability of the final chosen break, V_k, and previous unchosen breaks, $1 - V_l$, where each break probability is drawn from a $Beta$

distribution with parameters $1, \gamma$

$$\pi_k = V_k \prod_{l=1}^{k-1} (1 - V_l), \text{ with } V_k \sim Beta(1, \gamma).$$

The metaphor to stick braking is that we successively break off portions of a stick until we stop by sampling V_k. The longer we sample the shorter will be the remaining stick and the smaller the π_k. With respect to sampling clusters, this signifies that successive clusters fewer members (i.e. topics have successively lower frequencies).

The Chinese restaurant process (CRP) is another common metaphor used to explain sampling from a DP. We imagine a restaurant of infinite size with clients occupying tables and all clients at a table being served the same *dish* for that table. When the next client enters the restaurant, the client chooses to sit at a table in proportion to the number of clients already at the table with some small weight, γ, of sitting at an empty table.

The process is modeled as a sequential process on the last client entering the restaurant. The ϕ_i are the latent parameters and correspond to the the *dish* that the client is served; which is the same as ϕ_t^* the *dish* for the table at which the client sits. The distribution of latent variables for clients is given as:

$$\phi_{n+1} | \phi_1, \ldots, \phi_n \sim \frac{1}{\gamma + n} (\gamma H + \sum_{t=1}^{m} n_t \delta_{\phi_t^*}),$$

where n_t is the number of clients at table t, n is the total number of clients, and m is the number of occupied tables. I use ϕ_t^* for the dish at table t, which could be served at more than one table, reserving ϕ_k^{**} for unique dishes served for the entire restaurant.

The CRP gives equivalent results to the stick breaking construction. An advantage is that it translates fairly readily to Monte Carlo Markov chain sampling process. In topic analysis, clients correspond to observed values (e.g. words), and tables with their corresponding ϕ_k values to topic compositions.

Hierarchical Dirichlet Process Mixture Model Let's expand the concept of a DP model to hierarchical DP mixture model. Applications are multiple pictures, restaurants, production lines, or documents. Modeling uses "conditionally independent hierarchical models of grouped data" (**TehYee:2010a**), where *hierarchical* is used in the sense of probability distributions.

We shall use the DP model developed above, but now consider that observations are grouped in some manner (e.g. by document). DPs are allocated for each group where each group's DP is based on the same parent DP. The specific application is that of modeling flat topics over multiple documents and allowing for an undefined number of topics.

We observe groups indexed by j, where $j = 1, \dots, J$ and for each group we observe the collection of values $\{x_{j,1}, \dots, x_{j,n_j}\}$ with corresponding latent parameters $\{\phi_{j,1}, \dots, \phi_{j,n_j}\}$ where each $\phi_{j,i}$ is iid over the document specific DP, G_j, while each $x_{j,i}$ is distributed as $F(\phi_{j,i})$. This is summarized as

$$x_{j,i}|\phi_{j,i} \sim F(\phi_{j,i}),\ \phi_{j,i} \sim G_j,\ G_j|\alpha, G_0 \sim DP(\alpha, G_0),\ G_0|\gamma, H \sim DP(\gamma, H).$$

H is a continuous distribution; G_0 is a DP and is discrete with probability one. Each G_j is derived from G_0 and so inherits the atoms, $\delta_{\phi_k^{**}}$, from G_0.

In cluster analysis, observations over groups would still use the clusters designated by ϕ_k^{**} and the observations $x_{j,i}$ would be clustered without losing their group identity. In topic analysis of text, groups correspond to documents and the ϕ_k^{**} correspond to the multinomial probabilities of selecting word $x_{j,i}$ for the topic parameterized by ϕ_k^{**}.

Each group or document DP G_j can be written as:

$$G_j = \sum_{k=1}^{\infty} \pi_{j,k} \delta_{\phi_k^{**}}$$

where the ϕ_k^{**} are the unique parameters sampled from H via the atoms $\delta_{\phi_k^{**}}$ inherited from G_0, and the $\pi_{j,k}$ are weights specific to each group giving each G_j as a "reweighted sum of atoms in G_0" (**TehYee:2010b**). The weight vectors π_j are not independent from π as each G_j is sampled from $DP(\alpha, G_0)$. **TehYee:2010b** reports the relation of π_j as draws from a DP with concentration α and base distribution π, $\pi_{\mathbf{j}}|\alpha, \pi \sim DP(\alpha, \pi)$. The $\pi_{\mathbf{j}}$ will have the same expected value as the π and a greater variance.

Continuing with the Chinese restaurant metaphor, the process now becomes the Chinese restaurant franchise (CRF). Groups are represented as restaurants, clients enter restaurants and select a table as before with probability proportional to the number of clients at the table at that restaurant and sometimes choosing an empty table with probability proportional to α. Each table serves just one dish; if the client sits at an empty table, then a dish is assigned the table from the franchise menu with probability proportional to all the tables serving this dish in the franchise and sometimes serving an entirely new dish with likelihood proportional to γ.

The probability distribution for client i entering restaurant j being served dish $\phi_{j,i}$ is

$$\phi_{j,i}|\phi_{j,1}, \dots, \theta_{j,i-1}, \alpha, G_0 \sim \sum_{t=1}^{m_{j,\cdot}} \frac{n_{j,t,\cdot}}{\alpha + n_{j,\cdot,\cdot}} \delta_{\phi_{j,k}^{*}} + \frac{\alpha}{\alpha + n_{j,\cdot,\cdot}} G_0$$

where $\phi_{j,i}$ are dishes served to clients, α is the weight for selecting an empty table, G_0 is the base distribution of dishes across the restaurant franchise. If client i sits at table t of restaurant j, $t_{j,i}$, then the client receives dish $\phi_{j,i}$ which is the dish served at table $t_{j,i}$, and so $\phi_{j,i} = \phi^*_{j,t_{j,i}}$.

If the client sits at an empty table, then a new dish $\phi^*_{j,t_{j,i}}$ is selected from the franchise menu for the table as

$$\phi^*_{j,t}|\phi^*_{1,1},\ldots,\phi^*_{1,m_1},\ldots,\phi^*_{j,t-1},\gamma,H \sim \sum_{k=1}^{K} \frac{m_{\bullet,k}}{\gamma + m_{\bullet,\cdot}\delta_{\phi^{**}_k}} + \frac{\gamma}{\gamma + m_{\bullet,\cdot}} H$$

where $\phi^*_{j,t}$ are the dishes served at the tables of restaurant j, γ is the weight of selecting a dish not currently being served in the franchise, H is the base distribution of dishes, ϕ^{**}_k. If the dish being served at table t of restaurant j is $\phi^*_{j,t}$ comes from dish k of the franchise menu, then $\theta^*_{j,t} = \phi^{**}_k$. If a new dish is selected, it is selected according to the base distribution H.

This discussion of the hierarchical DP using the Chinese restaurant franchise (CRF) metaphor highlights the importance of group and unit (i.e. table) frequency distributions. In the CRF, restaurants are groups j, clients are observations $x_{j,i}$, seating at a table assigns the cluster parameterization which comes from the unique dishes of the franchise ϕ^{**}_k. Specifically, clients are words, dishes are topics, restaurants are documents. While topic compositions are the same across the corpus, topic mixtures vary by document.

Tree Structured Dirichlet Processes Let's expand the concept of a DP from above to a tree structure (i.e. a hierarchy) of DPs starting at the root. As in (**Paisley:2015**), refer to a particular topic by the path from the root to this topic as a vector $\mathbf{h} = \{h_0,\ldots,h_l\}$ where h_g references an element of the path by level in the topic tree. At each node of the topic tree, instantiate a DP. This model generalizes the DP model as follows. We observe a collection of values $\{x_1,\ldots,x_n\}$ with corresponding latent parameters $\{\phi_{\mathbf{h}_1},\ldots,\phi_{\mathbf{h}_n}\}$. The paths $\mathbf{h}_i$ denote particular nodes of the nested tree and each $\phi_{\mathbf{h}_i}$ an independent sequence of draws from the root down to the node at level l of the path $\mathbf{h}_i$.

There is a DP for each node of tree structure summarized as follows. Words x_i are sampled from a multinomial distribution $F(\phi_{\mathbf{h}[l]})$ parameterized by $\phi_{\mathbf{h}[l]}$ where $\phi_{\mathbf{h}[l]}$ is sampled from the parent DP $G_{\mathbf{h}[l-1]}$, and the parent DP is constructed as before from the concentration parameter γ and base distribution H. This is summarized as

$$x_i|\phi_{\mathbf{h}[l]} \sim F(\phi_{\mathbf{h}[l]}),\ \phi_{\mathbf{h}[l]}|G_{\mathbf{h}[l-1]} \sim G_{\mathbf{h}[l-1]},\ G_{\mathbf{h}[l-1]}|\alpha,H \sim DP(\gamma,H).$$

Express each $G_{\mathbf{h}[l]}$ as a weighted sum of atoms, $G_{\mathbf{h}[l]} = \sum_{k=1}^{\infty} \pi_{\mathbf{h}[l],k}\delta_{\phi^{**}_{\mathbf{h}[l],k}}$, based on the stick breaking construction and CRP as before. The result is a tree structure hierarchy of DPs where the distributions $F(\phi_{\mathbf{h}[l]})$ and $G_{\mathbf{h}[l-1]}$ are independent for each node.

Full Tree Structured and Hierarchical Model Let's combine the tree structured DPs and hierarchical DP mixture models to arrive at a model that handles grouped observations via conditionally independent hierarchical DP mixture models for a tree of nested DP mixture models.

There is a DP mixture model for each node of each document tree structure where the document tree structures nodes are derived from the global tree structure nodes sharing the same cluster parameters $\phi_{\mathbf{h}[l]}$ and deriving document node DPs from the corresponding global node DPs. Words $x_{j,i}$ are sampled from a multinomial distribution $F(\phi_{j,\mathbf{h}[l]})$ parameterized by $\phi_{j,\mathbf{h}[l]}$ where $\phi_{j,\mathbf{h}[l]}$ is sampled from the parent DP $G_{j,\mathbf{h}[l-1]}$, and the parent DP is constructed from the concentration parameter α and the corresponding DP of the global tree $G_{\mathbf{h}[l-1]}$. Finally, as we saw above, the DPs from the global tree are constructed from the concentration parameter γ and the base distribution H. The combined model is summarized as

$$\begin{aligned} x_{j,i}|\phi_{j,\mathbf{h}[l]} &\sim F(\phi_{j,\mathbf{h}[l]}), & \phi_{j,\mathbf{h}[l]}|G_{j,\mathbf{h}[l-1]} &\sim G_{j,\mathbf{h}[l-1]}, \\ G_{j,\mathbf{h}[l-1]}|\alpha, G_{\mathbf{h}[l-1]} &\sim DP(\alpha, G_{\mathbf{h}[l-1]}), & G_{\mathbf{h}[l-1]}|\gamma, H &\sim DP(\gamma, H). \end{aligned}$$

We can express each $G_{j,\mathbf{h}[l-1]}$ as reweighted sum of atoms in the corresponding global DP, $G_{j,\mathbf{h}[l-1]} = \sum_{k=1}^{\infty} \pi_{j,k}\delta_{\phi_k^{**}}$, where the δ_k^{**} are unique topics sampled from H and inherited from $G_{\mathbf{h}[l-1]}$, and the weights $\pi_{j,k}$ specific to group j give a reweighted sum over atoms δ_k^{**}. Even though there are multiple processes sampling from H, uniqueness is guaranteed in that H is continuous.