Nested Chinese Restaurant Franchise Processes: Applications to User Tracking and Document Modeling

Amr Ahmed AMRA@GOOGLE.COM
Research @ Google

Linagjie Hong LIANGJIE@YAHOO-INC.COM
Yahoo! Labs

Alexander J. Smola ALEX@SMOLA.ORG
Carnegie Mellon University

Abstract

Much natural data is hierarchical in nature. Moreover, this hierarchy is often shared between different instances. We introduce the nested Chinese Restaurant Franchise Process to obtain both hierarchical tree-structured representations for objects, akin to (but more general than) the nested Chinese Restaurant Process while sharing their structure akin to the Hierarchical Dirichlet Process.

Moreover, by decoupling the *structure generating* part of the process from the components responsible for the observations, we are able to apply the same statistical approach to a variety of user generated data. In particular, we model the joint distribution of microblogs and locations for Twitter for users. This leads to a 40% reduction in location uncertainty relative to the best previously published results. Moreover, we model documents from the NIPS papers dataset, obtaining excellent perplexity relative to (hierarchical) Pachinko allocation and LDA.

1. Introduction

Micro-blogging services such as Twitter, Tumblr and Weibo have become important tools for online users to share breaking news, interesting stories, and rich media content. Many such services now provide location services. That is, the messages have both textual and spatiotemporal information, thus the need to design models that account for both modalities jointly. It is reasonable to assume that there exists some degree of correlation between content and location and moreover, that this distribution be user-specific.

Likewise, longer documents contain a mix of topics, albeit not necessarily at the same level of differentiation between different topics. That is, some documents might address, amongst other aspects, the issue of computer science, albeit one of them in entire generality while another one might delve into very specific aspects of machine learning. As with microblogs, such data requires a hierarchical model that shares structure between documents and where the objects of interest themselves exhibit a rich structural representation (e.g. a distribution over a tree).

Such problems have attracted a fair amount of attention. For instance (Mei et al., 2006; Wang et al., 2007; Eisenstein et al., 2010; Cho et al., 2011; Cheng et al., 2011) all address the issue of modeling location and content in microblogs. Moreover, it has recently come to our attention (by private communication) that (Paisley et al., 2012) independently proposed a model similar to the nested Chinese Restaurant Franchise Process (nCRF) of this paper. The main difference is found in the inference algorithm (variational rather than collapsed sampling) and the different range of applications (documents rather than spatiotemporal data) as well as our parameter cascades over the tree.

Most related regarding microblogs is the work of Eisenstein et al. (2010; 2011); Hong et al. (2012). They take regional language variations and global topics into account by bridging *finite* mixture Gaussian models and topic models. These models usually employ a flat clustering model of locations. This flat structure is unnatural in terms of the language model: while it is reason-

Proceedings of the 30^{th} International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

able to assume that New Yorkers and San Franciscans might differ in terms of the content of the tweets, it is also reasonable to assume that American tweets, as a whole, are more similar to each other, than to tweets from Egypt, China or Germany. As a side effect, location prediction is not always satisfactory.

Key Contributions: We introduce a model that combines the advantages of the Hierarchical Dirichlet Process (HDP) of Teh et al. (2006) and the nested Chinese Restaurant Process (nCRP) of Blei et al. (2010) into a joint statistical model that allows each object to be represetned as a mixture of paths over a tree. This extends the hierarchical clustering approach of (Adams et al., 2010) and also includes aspects of Pachinko allocation (Mimno et al., 2007) as special cases. The model decouples the task of modeling hierarchical structure from that of modeling observations.

Moreover, we demonstrate in two applications, microblogs and NIPS documents, that the model is able to scale well and that it can provide highly accurate estimates in both cases. That is, for microblogs we obtain a location inference algorithm with significantly improved accuracy relative to the best previously published results. Moreover, for documents, we observe significant gains in perplexity.

2. Background

Bayesian nonparametrics is rich in structured and hierarchical models. Nonetheless, we found that no previously proposed structure a good fit for the following key problem when modeling tweets: we want to model each user's tweets in a hierarchical tree-like structure akin to the one described by nested Chinese Restaurant Process or the hierarchical Pachinko Allocation Model. At the same time we want to ensure that we have sharing of statistical strength *between* different users' activities by sharing the hierarchical structure and the associated emissions model. For the purpose of clarity we briefly review the two main constituents of our model — HDP and nCRP.

Franchises and Hierarchies: A key ingredient for building hierarchical models is the Hierarchical Dirichlet Process (HDP). It is obtained by coupling draws from a Dirichlet process by having the reference measure itself arise from a Dirichlet process (Teh et al., 2006). In other words, rather than drawing the distribution G from a Dirichlet process via $G \sim \mathrm{DP}(H, \gamma)$ (for the underlying measure H and concentration parameter γ) we now have

$$G_i \sim \mathrm{DP}(G_0, \gamma') \text{ and } G_0 \sim \mathrm{DP}(H, \gamma). \tag{1}$$

Here γ and γ' are appropriate concentration parameters. This means that we first draw atoms from H to obtain G_0. This is then, in turn, used as reference measure to obtain the measures G_i. They are discrete and share, by construction, atoms via G_0.

The Hierarchical Dirichlet Process is widely used in applications where different groups of data points would share the same settings of partitions, such as (Teh et al., 2006; Beal et al., 2002). In the context of document modeling the HDP is used to model each document as a DP while sharing the set of atoms (mixtures or topics) across all documents. This is precisely what we also want when assessing distributions over trees — we want to ensure that the (partial) trees attached to each user share attributes among all users.

Integrating out all random measures, we arrive as what is known as the Chinese Restaurant Franchise (CRF). In this metaphor each restaurant maintains its set of tables but shares the same set of mixtures. A customer at restaurant k can chose to sit at an existing table with a probability proportional to the number of customers sitting on this table, or start a new table with probability α and chose its dish from a global distribution.

The Nested Chinese Restaurant Process: CRPs and CRFs allow objects, such as documents, to be generated from a single mixture (topic). However, they do not provide a relationship between topics. One option to address this issue is to introduce a tree-wise dependency. This was proposed in the nested Chinese Restaurant Process (nCRP) by (Blei et al., 2010). It defines an infinite hierarchy, both in terms of width and depth. In the nCRP, a set of topics (mixtures) are arranged over a tree-like structure whose semantic is such that parent topics are more general than the topics represented by their children. A document in this process is defined as a path over the tree, and it is generated from the topics along that path using an LDA-like model. In particular, each node in the tree defines a Chinese Restaurant Process over its children. Thus a path is defined by the set of decisions taken at each node. While this provides more expressive modeling, it still only allows each document to have a single path over the tree – a limitation we overcome below.

3. Nested Chinese Restaurant Franchise

We now introduce the Nested Chinese Restaurant Franchise (nCRF). As its name suggests, it borrows both from the Chinese Restaurant Franchise, allowing us to share strength between groups, and the Nested Chinese Restaurant Process, providing us with a hierarchical distribution over observations.

For clarity of exposition and concepts we will distinguish between the structure generating nCRF and the process generating observations from a hierarchical generative model, once the structure variable has been determined. This is beneficial since the observation space can be rather vast and structured.

3.1. Basic Idea

The nested Chinese Restaurant Process (nCRP) (Blei et al., 2010) provides a convenient way to impose a distribution over tree-like structures. However, it lacks a mechanism for 'personalizing' them to different partitions and restricts each object to only select a single path over the tree. On the other hand, franchises allow for such personalization (Teh et al., 2006), however they lack the hierarchical structure. We combine both . In keeping with one of the applications, the analysis of microblogs, we will refer to each partition requiring personalization as a *user*.

The basic idea is as follows: each user has its own tree-wise distribution, but the set of nodes in the trees, and their structure, such as parent-child relationships, are shared across all users in a franchise, as illustrated in Figure 1. Each node in all processes (global and user processes) defines a distribution over its children. This distribution is represented by the histograms attached to the vertices A, A_1, A_2 and B, B_1, B_2 respectively. A user first selects a node. Subsequently the generative model for the data associated with this particular vertex is invoked. For instance, user 1 first selects a sibling of node A_1 based on the local distribution or with probability proportional to α he creates a new child. In the latter case the child is sampled according to the global distribution associated with node A. Then user A continues the process until a path is fully created. For instance, if the selected node is B_1 then the process continues similarity. Thus Nodes A, A_1 and A_2 constitute a CRF process. In general, isomorphic nodes in the global and user processes are linked via a CRF process. Since the user selects a path by descending the tree, we call this the nCRF process. Equivalently data could be represented as an nHDP.

3.2. A Chinese Restaurant Metaphor

To make matters more concrete and amenable to sampling from the process we resort to a Chinese Restaurant metaphor. Consider the case where we want to generate a path for an observation generated by user u. We first start at the root node in the process of user u. This root node defines a CRP process over its children. Thus we can select an existing child or create a new child. In the later case, the global CRP associated with the root node is consulted. A child in the global tree is selected with probability proportional to its global usage across all users. Alternatively a new child node is created and thus made accessible to all other users.

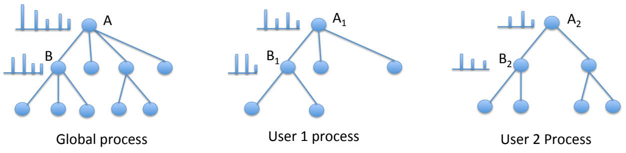

Figure 1. The nested Chinese Restaurant Franchise involving a common tree over components (left) and two subtrees representing processes for two separate subgroups (e.g. users). Each user samples from his own distribution over topics, smoothed by the global process. Thus each user process represents a nested Chinese Restaurant Process. All of them are combined into a common franchise.

All selection probabilities are governed using the standard CRF's self-reinforcing Dirichlet process mechanism. Note that we could equally well employ the strategy of Pitman & Yor (1997) in order to obtain power-law size distribution of the partitions. This is omitted for clarity of exposition. Once a child node is selected, the process recurses with that node until a full path is defined. We need some notation, as described in the table below:

v, w, r	denotes vertices (nodes) in the tree
$\pi(v)$	parent of v
$\pi^i(v)$	i^{th} ancestor of v, where $\pi^0(v) = v$
$L(r)$	depth of vertex r in the tree, L(root)=0
$C(v)$	children of v
$C^u(v)$	children of v for user u, note that $C^u(v) \in C^(v)$
n_v^u	occurrences of v in user u's process
n_v	occurrences of v in the global process

Moreover, for convenience, in order to denote the *path* explicitly, we use its end vertex in the tree (since a path from the root is uniquely determined by its end vertex). Moreover, we will denote by $n_{vw} := n_w$ and by $n_{vw}^u := n_w^u$ the counts for a specific child of v (this holds trivially since children only have a single parent). This now allows us to specify the collapsed generative probabilities at vertex v. The probability of selecting an existing child is

$$\Pr\{v \to w\} = \begin{cases} \frac{n_{vw}^u}{n_v^u + \alpha} & \text{if } w \in C^u(v) \text{ from user tree} \\ \frac{\alpha}{n_v^u + \alpha} & \text{otherwise} \end{cases}$$

Whenever we choose a child not arising from the user tree we fall back to the distribution over the common tree. That is, we sample as follows

$$\Pr\{v \to w\} = \begin{cases} \frac{n_{vw}}{n_v + \beta} & \text{if } w \in C(v) \\ \frac{\beta}{n_v + \beta} & \text{if this is a new child} \end{cases}$$

Combining both cases we have the full probability as

$$\Pr\{v \to w\} = \begin{cases} \frac{n^u_{vw}}{n^u_v+\alpha} + \frac{\alpha}{n^u_v+\alpha}\frac{n_{vw}}{n_v+\beta} & \text{if } w \in C^u(v) \\ \frac{\alpha}{n^u_v+\alpha}\frac{n_{vw}}{n_v+\beta} & \text{if } w \in C(v)\backslash C^u(v) \\ \frac{\alpha}{n^u_v+\alpha}\frac{\beta}{n_v+\beta} & \text{if } w \notin C(v) \end{cases}$$

Note here that we used a direct-assignment representation for the CRF at each node to avoid overloading notation by maintaing different tables for the same child at each node (see (Teh et al., 2006) for an example). This is correct due to the coagulation/fragmentation equivalence in Dirichlet Processes derived by (James, 2010). In other words, in all CRPs, each child node is represented by a single table, hence table and child become synonymous and we omit the notion of tables. For inference, an axillary variable method is used to link the local n^u_{ij} and global counts n_{ij} using the Antoniak distribution as described by (Teh et al., 2006).

3.3. Termination and observation process

Once a child node is selected, the process is repeated until a full path is defined. To ensure finite paths we need to allow for the probability of termination at a vertex. This is achieved in complete analogy to (Adams et al., 2010), that is, we treat the probability of terminating at a vertex in complete analogy to that of generating a special child. Note that as in hierarchical clustering we could assign a different smoothing prior to this event (we omit details for clarity). In terms of count variables we denote this by

$$n_{v0} := n_v - \sum_{w \in C(v)} n_w \text{ and } n^u_{v0} := n^u_v - \sum_{w \in C^u(v)} n^u_w$$

In other words, termination at a given node behaves just like yet another child of the node. All probabilities as discussed above for $\Pr\{v \to w\}$ hold analogously.

The above process defines a prior over trees where each objects can chose multiple paths over the tree. To combine this with a likelihood model , we need to address the observation model. We postulate that each node v in the tree is associated with a distribution ψ_v. To leverage the tree structure, we cascade these distributions over the tree such that ψ_r and $\psi_{\pi(r)}$ are similar. The specific choice of such a distribution depends on the nature of the parameter: we use a Dirichlet-multinomial cascade for discrete attributes and Gaussian cascades for continuous attributes. In the following section we will give two application of nCRF in modeling user locations from geotagged microblogs and in modeling topic hierarchy from a document collection producing a non-parametric version of the successful hPAM model (Mimno et al., 2007).

4. Generating Microblogs with nCRFs

To illustrate the effect of our model we describe how to model microblogs using nCRFs. We are given collections of tweets with timestamps, location information, and information regarding the author of the tweets. We want to use this to form a joint generative model of both location and content. Rather importantly, we want to be able to capture both the relation between locations and content while simultaneously addressing the fact that different users might have rather different profiles of location affinity. With some slight abuse of terminology we will use tweets and documents interchangeably to mean the same object in this section.

We aim to arrange content and location preferences in a tree. That is, we will assume that locations drawn from the leaves of a vertex are more similar between each other than on another vertex of the tree. Likewise, we assume hierarchical dependence of the language model, both in terms of content of the region specific language models and also in terms of prevalence of global topics. Secondly we assume that users only select a subtree of the global topic and location distribution and generate news based on this. By intertwining location and topical distributions into a *joint* model we are able to dynamically trade off between improved spatial accuracy and content description.

We use the nCRF process to model this problem. Here each object is a user u and elements inside each object are tweets denoted as d. Each node in the hierarchy denotes a region. We associate with each element (tweet d) a latent region r and a set of hidden variables z that would become apparent shortly. We assume that there exist a set of T global background topics. We denote each global topic by $\Pi_i \sim \mathrm{Dir}(\eta)$. To complete the generative process we need to specify the parameters associated with each node in the tree. We let $\psi_r = (\mu_r, \Sigma_r, \phi_r, \theta_r)$ corresponding to: the region mean and covariance, the region's language model and the region's topic mixing vector respectively.

Hierarchical location model: We consider a hierarchical multivariate Gaussian model in analogy to (Adams et al., 2010). The main distinction is that we need not instantiate a shrinkage step towards the origin at each iteration. Instead, we simply assume an additive Gaussian model. We are able to achieve this since we assume decreasing variance when traversing the hierarchy (Adams et al. (2010) did not impose such a constraint).

$$\mu_r \sim \mathcal{N}\left(\mu_{\pi(r)}, \Sigma_{\pi(r)}\right) \text{ and } \Sigma_r = L^{-1}(r)\Sigma_0. \quad (2)$$

Here Σ_0 is the covariance matrix of the root node. In other words, we obtain a tree structured Gaussian Markov Random Field. This is desirable since inference in it is fully tractable in linear time by means of message passing.

Location specific language model (ϕ): Using the intuition that geographical proximity is a good prior for similarity in a location specific language model we use a hierarchical Dirichlet Process to capture such correlations. In other words, we draw the root-level language model from

$$\phi_0 \sim \mathrm{Dir}(\eta). \tag{3}$$

At lower levels the language model is drawn using the parent language model as a prior. That is

$$\phi_i \sim \mathrm{Dir}\left(\omega\phi_{\pi(r)}\right) \tag{4}$$

In doing so, we will obtain more specific topics at lower levels whereas at higher levels less characteristic tokens are more prevalent.

Location specific mix of topics (θ_r): To model hierarchical *distributions* over topics we can use a similar construction. This acts as a mechanism for mixing larger sets of words efficiently rather than just reweighting individual words. θ_r is constructed in complete analogy to the location specific language model. That is, we assume the hierarchical model

$$\theta_0 \sim \mathrm{Dir}(\beta) \text{ and } \theta_r \sim \mathrm{Dir}\left(\lambda\theta_{\pi(r)}\right) \tag{5}$$

After selecting a geographical region r for the tweet we generate the tweet from $T+1$ topics, $\overline{\Pi}$, using a standard LDA process, where the first T topics are the background language models and the $T+1^{th}$ topic is ϕ_r. The mixing proportion is governed by θ_r. Putting everything together, the generative process is

For each tweet d written by each user u:

(a) Sample a node $r_d \sim \mathrm{nCRF}(\gamma, \alpha, u)$.
(b) If node r_d is a *globally* new node then
   i. $\mu_{r_d} \sim \mathcal{N}\left(\mu_{\pi(r_d)}, \Sigma_{\pi(r_d)}\right)$
   ii. $\phi_{r_d} \sim \mathrm{Dir}\left(\omega\phi_{\pi(r_d)}\right)$
   iii. $\theta_{r_d} \sim \mathrm{Dir}\left(\lambda\theta_{\pi(r_d)}\right)$
(c) Sample a location $l_d \sim \mathcal{N}(\mu_{r_d}, \Sigma_{r_d})$.
(d) For each word $w_{(d,i)}$:
   i. Sample a topic index $z_{(d,i)} \sim \mathrm{Multi}(\theta_{r_d})$.
   ii. Sample word $w_{(d,i)} \sim \mathrm{Multi}(\overline{\Pi}_{z_{(d,i)}})$.

5. Modeling Documents with nCRFs

When applying nCRFs to document modeling the abstraction is slightly different. Now documents are the key reference unit. They are endowed with a tree-distribution over topics that generate words. Moreover, these distributions are then tied together in a franchise as discussed previously. Previous work such as the nCRP (Blei et al., 2010), PAM (Li & McCallum, 2006) and hPAM (Mimno et al., 2007) arrange the topics in a tree-like structure where the tree structure is fixed a-priori as in PAM and hPAM or learned from data as in nCRP. Moreover, in nCRP and PAM only leaf topics can emit words while in the other models both leaf topics and internal topics can emit words. Along the other dimension models such as PAM and hPAM allow each document to be represented as multiple paths over the tree while in nCRP each document is represented as a single path over the tree. However, only the nCRF can simultaneously learn the tree structure and allow each document to be represented as multiple paths over the tree:

For each word i in document d:

(a) Sample a node $v_{(d,i)} \sim \mathrm{nCRF}(\gamma, \alpha, d)$.
(b) If node $v_{d,i}$ is a *globally* new node then
   i. $\phi_{v_{(d,i)}} \sim \mathrm{Dir}\left(\omega\phi_{\pi(v_{(d,i)})}\right)$
(c) Sample word $w_{(d,i)} \sim \mathrm{Multi}(\phi_{v_{(d,i)}})$.

In the above model each node v in the tree represents a topic and we endow each node with a multinomial distribution over the vocabulary. This model constitutes a non-parametric version of the hPAM model in which the tree structure is learned from data.

6. Inference

Below we describe the generic aspects of the inference algorithm for nCRFs. Model specific aspects are relegated to the appendix. Given a set of objects $\mathbf{x}_{1:N}$ where object $\mathbf{x}_o = (x_{o1}, x_{o2}, \cdots, x_{o|x|})$, the inference task is to find the posterior distribution over: the tree structure, each object's distribution over the tree, node assignments r_{oj} to each element x_{oj}, additional hidden variables associated with each element z_{oj}, and the posterior distribution over the cascading parameters ψ. For example, in microblogs an object corresponds to a user, each element x_{oj} is a tweet (which is itself is a bag of words and a location), and z_{oj} is a set of topic indicators for words in tweet x_{oj}. In document modeling each object is a document which is composed of a bag of words — z is empty in this application.

We construct a Markov chain over $(\mathbf{r}, \mathbf{z}, \psi)$ and we alternate sampling each of them from their conditional distributions. We first sample the node assignment r_{oj} followed by sampling z_{oj} if needed, then after a full sweep over the data, we sample Ψ (the latter greatly simplifies sampling from the observations model). We give two algorithms for sampling r_{oj}: an exact Gibbs Sampling algorithm and an approximate Metropolis Hastings method that utilizes a level-wise proposal. We briefly discuss sampling $(\mathbf{z}, \psi)$ deferring the details to the appendix as they depend on the application.

6.1. Exact Gibbs Sampling of r_{oj}

The conditional probability for generating element x_{oj} from node r_{oj} is given by

$$\begin{aligned} & P(r_{oj} = r | x_{oj}, z_{oj}, \text{rest}) \\ \propto \ & P(r_{oj} = r|\text{rest}) p(x_{oj}, z_{oj} | r_{oj} = r, \text{rest}) \end{aligned} \tag{6}$$

where the prior is given by

$$P(r_{oj} = r|\text{rest}) \propto \prod_{i=0}^{l(r)-1} P(\pi^{i+1}(r) \rightarrow \pi^{i}(r))$$

In it, each component of this product is given by the nCRF process. In other words, the product just computes the node selection probabilities along the path from the root to node r. Note that for a tree with n nodes, we need to consider $2n$ outcomes, since we can add a child to every existing node. The second component is the likelhood model (it depends on the application). The complexity of this algorithm is $O(dn)$ where d is the depth of the tree. A better approach uses dynamic programming to reduce sampling complexity to $O(n)$. This holds since the probabilities are given by a product that we can evaluate iteratively.

6.2. Metropolis-Hasting Sampling of r_{oj}

Instead of sampling a path for element x_{oi} as a block by sampling its end node, we use a level-wise strategy to sample a latent node at each level until we hit an existing node (i.e. child 0). Starting from the root node, assume that we reached node v on the tree, then we can descend the tree as follows:

1. Stay on the current node – i.e. pick child 0.
2. Move to a child node w of v other than child 0.
3. Create a new child of node v and move to it.

Assume we ended with $r_{oi} = r$. The path from the root to node r is thus given by: $(\pi^{L(r)}, \cdots, \pi^{0}(r))$. Clearly this procedure gives an approximate conditional probability to sampling a node assignment, therefore, we consider it as a proposal distribution whose form is given by:

$$q(r) = \prod_{i=0}^{l(r)-1} \frac{\mathrm{P}\big(\pi^{i+1}(r) \rightarrow \pi^{i}(r)\big) p(x_{oi}, z_{oi} | \pi^{i}(r))}{\sum_{r' \in C(\pi^{i+1}(r))} \mathrm{P}\big(\pi^{i+1}(r) \rightarrow r'\big) p(x_{oi}, z_{oi} | r')}$$

Here the selection probabilities are as given by the nCRF process. We accept a node assignment r^{new} generated by this proposal to replace an existing assignment r^{old} with probability s:

$$s = \min\left(1, \frac{q(r^{\text{old}}) P(r^{\text{new}}|\text{rest})}{q(r^{\text{new}}) P(r^{\text{old}}|\text{rest})}\right)$$

Note that $P()$ is *proportional* to the exact probability as in (6), however only evaluated *proportionally* at the old and proposed node assignments. The complexity of this algorithm is O(dC), where d is the depth of the tree and C is the average number of children per node.

6.3. Sampling $(\mathbf{z}, \psi)$

Sampling the variables in Ψ depends on the type of the cascade. We do not collapse variables corresponding to a Gaussian-Gaussian cascade, and as such to sample them, we compute the posterior over the cascade using a Multi-scale kalman filtering algorithm and then sample from this posterior (see Appendix for more details). We collapse variables corresponding to a Dirichlet-Multinomial cascade and we use an auxiliary variable method similar to (Teh et al., 2006) to sample them either using the Antoniak distribution for cascaded over variables with moderate cardinality (as in the topic distributions), or using the min-path/max-path approximation for cascades over variables with large cardinalities (as in the topic word distributions). We details each of these pieces in the Appendix for lack of space. Moreover, the use of auxiliary variables allows for efficient computation of the likelihood component $P(x_{oj}, z_{oj}|r_{oj})$ as it decouples nodes across various levels of the tree.

Sampling z, if required, depends on the application. For the twitter application the equations are the same as in a standard LDA (with proper tree-based smoothing – see Appendix). Finally computing the data likelihood component $P(x_{oj}, z_{oj}|r_{oj})$ is straightforward in the document modeling application. For the twitter application this term factors as: $P(w_d|r_d, z_d)P(z_d|r_d)P(l_d|r_d)$. The first term reduces to only computing the probability of generating the words associated with a regional language model since the probability of the rest of the words does not depend on the node. This distribution amounts to a standards ratio of two log-partition functions. Similarly computing $P(z_d|r_d)$ reduces to the ratio for two log-partition functions, and $p(l_d|r_d)$ just follows a MVN distribution (As we don't collapse the continuous variables).

7. EXPERIMENTS

7.1. User Location modeling

We demonstrate the efficacy of our model on two datasets obtained from Twitter streams. Each tweet contains a real-valued latitude and longitude vector. We remove all non-English tweets and randomly sample 10,000 Twitter users from a larger dataset, with their full set of tweets between January 2011 and May 2011, resulting 573,203 distinct tweets. The size of

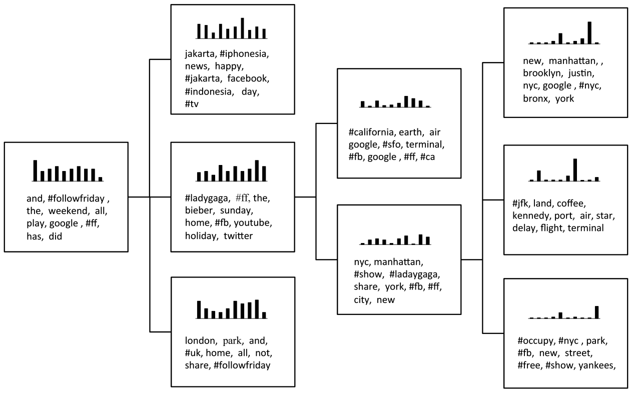
and, #followfriday , the, weekend, all, play, google , #ff, has, did
jakarta, #iphonesia, news, happy, #jakarta, facebook, #indonesia, day, #tv
#ladygaga, #ff, the, bieber, sunday, home, #fb, youtube, holiday, twitter
london, park, and, #uk, home, all, not, share, #followfriday
#california, earth, air google, #sfo, terminal, #fb, google , #ff, #ca
nyc, manhattan, #show, #ladaygaga, share, york, #fb, #ff, city, new
new, manhattan, , brooklyn, justin, nyc, google , #nyc, bronx, york
#jfk, land, coffee, kennedy, port, air, star, delay, flight, terminal
#occupy, #nyc , park, #fb, new, street, #free, #show, yankees,

Figure 2. Portion of the tree structure discovered from DS1.

Table 1. Top ranked terms for some global topics.

Entertainment
video gaga tonight album music playing artist video
itunes apple produced bieber #bieber lol new songs
Sports
winner yankees kobe nba austin weekend giants
horse #nba college victory win
Politics
tsunami election #egypt middle eu japan egypt
tunisia obama afghanistan russian
Technology
iphone wifi apple google ipad mobile app online
flash android apps phone data

Table 2. Location accuracy on DS1 and DS2.

Results on DS1	**Avg. Error**	**Regions**
(Yin et al., 2011)	150.06	400
(Hong et al., 2012)	118.96	1000
Approx.	91.47	2254
MH	90.83	2196
Exact	83.72	2051

Results on DS2	**Avg. Error**	**Regions**
(Eisenstein et al., 2010)	494	-
(Wing & Baldridge, 2011)	479	-
(Eisenstein et al., 2011)	501	-
(Hong et al., 2012)	373	100
Approx.	298	836
MH	299	814
Exact	275	823

dataset is significantly larger than the ones used in some similar studies (e.g, (Eisenstein et al., 2010; Yin et al., 2011)). We denote this dataset as DS1. For this dataset, we split the users (with all her tweets) into **disjoint** training and test subsets such that users in the training set **do not** appear in the test set. In other words, users in the test set are like *new* users. This is the most adversarial setting. In order to compare with other location prediction methods, we also apply our model a dataset available at http://www.ark.cs.cmu.edu/GeoText/, denoted as DS2, using the same split as in (Eisenstein et al., 2010). (more analysis is given in Appendix B and in (Ahmed et al., 2013a))

Table 3. Accuracy of different approximations and sampling methods for computing ϕ_r.

Method	DS1	DS2
Minimal Paths	91.47	298.15
Maximal Paths	90.39	295.72
Antoniak	88.56	291.14

Table 4. Ablation study of our model

Results on DS1	**Avg. Error**	**Regions**
(Hong et al., 2012)	118.96	1000
No Hierarchy	122.43	1377
No Regional Language Models	109.55	2186
No Personalization	98.19	2034
Full Model.	91.47	2254

Results on DS2	**Avg. Error**	**Regions**
(Hong et al., 2012)	372.99	100
No Hierarchy	404.26	116
No Regional Language Models	345.18	798
No Personalization	310.35	770
Full Model.	298.15	836

Figure 2 provides a small *subtree* of the hierarchy discovered on DS1 with the number of topics fixed to 10. Each box represents a region where the root node is the leftmost node. The bar charts demonstrate overall topic proportions. The words attached to each box are the top ranked terms in regional language models (they are all in English since we removed all other content). Because of cascading patterns defined in the model, it is clear that topic proportions become increasingly sparse as the level of nodes increases. This is desirable as we can see that nodes in higher level represent broader regions. The first level roughly corresponds to Indonesia, the USA and the UK, under USA, the model discovers CA and NYC and then under NYC it discovers attraction regions. We show some global topics in Table 1 as well which are more generic than the regional language models.

7.1.1. Location Prediction

We test the accuracy by estimating locations for each tweet based on its content and the author (we repeat that train and test users are disjoint). For each new tweet, we predict its location as $\hat{l}_d$. We calculate the Euclidean distance between predicted value and the true location and average them over the whole test set $\frac{1}{N}\sum l(\hat{l}_d, l_d)$ where $l(a, b)$ is the distance and N is the total number of tweets in the test set. The average error is calculated in kilometres. We use three inference algorithms for our model here: 1) exact algorithm denoted as Exact, 2) M-H sampling, denoted as MH and 3) Approximation algorithm Approx which is the same an the M-H algorithm but always accepts the proposal.

For DS1 we compare our model with (Yin et al., 2011) and the state of the art algorithm in (Hong et al., 2012) that utilizes a sparse additive generative model to incorporate a background language models, regional language models and global topics and considers users' preferences over topics and *flat* fixed number of regions

For all these models, the prediction is done by two steps: 1) choosing the region index that can maximize the test tweet likelihood, and 2) use the mean location of the region as the predicted location. For `Yin 2011` and `Hong 2012`, the regions are the optimal regions which achieve the best performance. For our method, the error is calculated as the average of number of regions from several iterations after the inference algorithm converges. The results are shown in the top part of Table 2. As evident from this Table, our model peaks at a much larger number of regions than the number of regions corresponding to the best baseline models. We conjecture that this is due to the fact that the model organizes regions in a tree-like structure and therefore more regions are needed to represent the fine scale of locations. Moreover, the Approx and MH algorithm performs reasonably well compared to the Exact algorithm since cascading distributions over the tree helps constraint the model.

For `DS2` dataset, we compare against all the algorithms published on this dataset. Both (Eisenstein et al., 2010) and (Eisenstein et al., 2011) use a sparse additive models with different learning algorithms. All methods compared against assume a fixed number of regions and we report the best result from their papers (along with the best number of regions if provided). As evident from Table 2, we have approximately 40% improvement over the best known algorithm (Hong et al., 2012) (note that area accuracy is quadratic in the distance). Recall that all prior methods used a flat clustering approach to locations. Thus, it is possible that the hierarchical structure learned from the data helps the model to perform better on the prediction task.

7.1.2. Ablation Study

We compared the different methods used to sample the regional language model: the two approximate methods (min-path and max-path – See Appendix A) and the exact method of directly sampling from the Antoniak distribution based on `Approx`. As shown in Table 3. We can see that all three methods achieve comparable results although sampling using Antoniak distribution can have slightly better predictive results. However, it takes substantially more time to draw from the Antoniak distribution, compared to Minimal Paths and Maximal Paths. In Table 2, we only report the results by using Minimal Paths. Moreover, we investigated the effectiveness of different components of the model in terms of location prediction. We compare different variants of the model by removing one component of the model at a time. As shown in Table 4 each component enhances the result, however, the hierarchical component seems to be a key to the superior performance of our model. We compare in this table against state of the art results in (Hong et al., 2012).

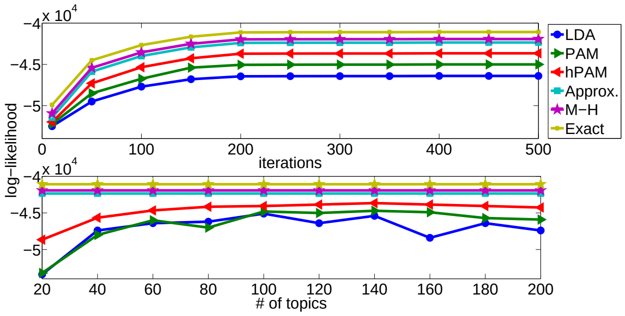

Figure 3. Performance on the NIPS data.

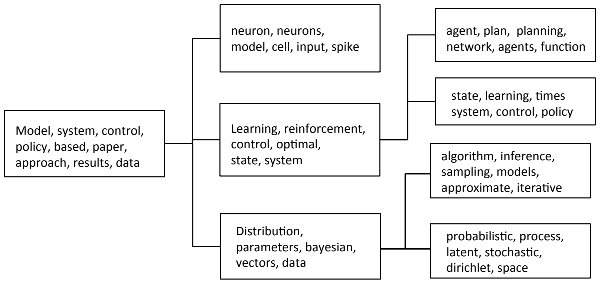
Model, system, control, policy, based, paper, approach, results, data
neuron, neurons, model, cell, input, spike
Learning, reinforcement, control, optimal, state, system
Distribution, parameters, bayesian, vectors, data
agent, plan, planning, network, agents, function
state, learning, times system, control, policy
algorithm, inference, sampling, models, approximate, iterative
probabilistic, process, latent, stochastic, dirichlet, space

Figure 4. Portion of the tree learnt from the NIPS dataset.

7.2. Document Modeling

We use the NIPS abstract dataset (NIPS00-12), which includes 1647 documents, a vocabulary of 11,708 words and 114,142 word tokens. We compare our results against PAM, hPAM and LDA (we omitted nCRP as it was shown in (Mimno et al., 2007) that hPAM outperformed it). We use the evaluation method from (Wallach et al., 2009) to evaluate the likelihood on held-out documents. As shown in Figure 3 our model outperforms the sate of the art hPAM model and the recent model by (Kim et al., 2010) which is equivalent to `Approx`. As we noted earlier our model can be regarded as a non-parametric version of hPAM. Figure 4 depicts a small portion of the tree.

8. Conclusion

In this paper we presented a modular approach to analyzing structured data. It allows us to model both the hierarchical structure of content, the hierarchical dependence between instances, and the (possibly) hierarchical structure of the observation generating process, all in one joint model. For future work, we plan to exploit distributed sampling techniques and data layout as in (Ahmed et al., 2012a; 2013b) in addition to hash-based sampling (Ahmed et al., 2012b) to scale the inference algorithm.

References

Adams, R., Ghahramani, Z., and Jordan, M. Tree-structured stick breaking for hierarchical data. In *NIPS*, pp. 19–27, 2010.

Ahmed, A., Aly, M., Gonzalez, J., Narayanamurthy, S., and Smola, A. Scalable inference in latent variable models. In *WSDM*, 2012.

Ahmed, A., Ravi, S., Narayanamurthy, S., and Smola, A. Fastex: Hash clustering with exponential families. In *NIPS*, 2012.

Ahmed,A., Hong, L., and Smola, A. Hierarchical Geographical Modeling of User locations from Social Media Posts. In *WWW*, 2013.

Ahmed,A., Shervashidze, N., Narayanamurthy, S., Josifovski, V., and Smola, A. Distributed large-scale natrual graph factorization. In *WWW*, 2013.

Beal, M. J., Ghahramani, Z., and Rasmussen, C. E. The infinite hidden markov model. In *NIPS*, 2002.

Blei, D., Ng, A., and Jordan, M. Latent dirichlet allocation. In *NIPS*, MIT Press, 2002

Blei, D., Griffiths, T., and Jordan, M. The nested chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM*, 57(2):1–30, 2010.

Cheng, Z., Caverlee, J., Lee, K., and Sui, D. Exploring millions of footprints in location sharing services. In *ICWSM*, 2011.

Cho, E., Myers, S. A., and Leskovec, J. Friendship and mobility: user movement in location-based social networks. In *KDD*, pp. 1082–1090, New York, NY, USA, 2011. ACM.

Chou, K.C., Willsky, A.S., and Benveniste, A. Multiscale recursive estimation, data fusion, and regularization. *IEEE Transactions on Automatic Control*, 39(3):464 –478, mar 1994.

Cowans, P. J. *Probabilistic Document Modelling*. PhD thesis, University of Cambridge, 2006.

Eisenstein, J., O'Connor, B., Smith, N. A., and Xing, E.P. A latent variable model for geographic lexical variation. In *Empirical Methods in Natural Language Processing*, pp. 1277–1287, 2010.

Eisenstein, J., Ahmed, A., and Xing, E. Sparse additive generative models of text. In *International Conference on Machine Learning*, pp. 1041–1048, New York, NY, USA, 2011. ACM.

Hong, L., Ahmed, A., Gurumurthy, S., Smola, A., and Tsioutsiouliklis, K. Discovering geographical topics in the twitter stream. In *World Wide Web*, 2012.

James, L. F. Coag-frag duality for a class of stable poisson-kingman mixtures, 2010. URL `http://arxiv.org/abs/1008.2420`.

Kim, J., Kim, D., Kim, S., and Oh, A. Modeling Topic Hierarchies with the Recursive Chinese Restaurant Process. In *CIKM*, , 2012.

Li, W. and McCallum, A. Pachinko allocation: Dag-structured mixture models of topic correlations. In *ICML*, 2006.

Li, W., Blei, D., and McCallum, A. Nonparametric bayes pachinko allocation. In *UAI*, 2007.

Mei, Q., Liu, C., Su, H., and Zhai, C.X. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proceedings of WWW*, pp. 533–542, New York, NY, USA, 2006. ACM.

Mimno, D.M., Li, W., and McCallum, A. Mixtures of hierarchical topics with pachinko allocation. In *ICML*, volume 227, pp. 633–640. ACM, 2007.

Paisley, J., Wang, C., Blei, D., and Jordan, M. I. Nested hierarchical dirichlet processes. Technical report, 2012. http://arXiv.org/abs/1210.6738.

Pitman, J. and Yor, M. The two-parameter poisson-dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25(2):855–900, 1997.

Teh, Y., Jordan, M., Beal, M., and Blei, D. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(576):1566–1581, 2006.

Wallach, H. *Structured Topic Models for Language*. PhD thesis, University of Cambridge, 2008.

Wallach, Hanna M., Murray, Iain, Salakhutdinov, Ruslan, and Mimno, David. Evaluation methods for topic models. In *ICML*, 2009.

Wang, C., Wang, J., Xing, X., and Ma, W.Y. Mining geographic knowledge using location aware topic model. In *Proceedings of the 4th ACM workshop on Geographical Information Retrieval*, pp. 65–70, New York, NY, USA, 2007. ACM.

Wing, B.P. and Baldridge, J. Simple supervised document geolocation with geodesic grids. In *Proceedings of ACL*, 2011.

Yin, Z., Cao, L., Han, J., Zhai, C., and Huang, T. Geographical topic discovery and comparison. In *World Wide Web*, pp. 247–256, New York, NY, USA, 2011. ACM.

Supplementary Material: Nested Chinese Restaurant Franchise Process Applications to User Tracking and Document Modeling

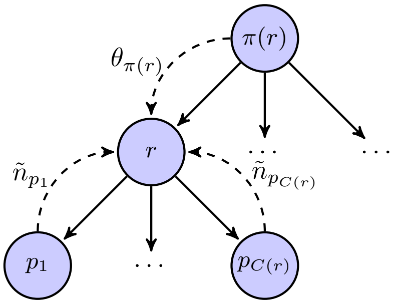

Figure 5. This is a demonstration of sampling θ_r, the distribution over topics for node r. The sampling is drawn from a Dirichlet distribution with parameters consisting of count statistics n_r from node r, pseudo counts $\tilde{n}_r$ gathering from its children nodes and topic proportions $\theta_{\pi(r)}$ from its parent node.

APPENDIX A: Inference in the Twitter Model

In this Section, we detail the sampling equations for $(\mathbf{z}, \mathbf{\Psi})$ in the twitter application for concreteness.

A.1 Sampling Topic Proportions

Since topic proportions for different regions are linked through the cascading process defined in Equation (5), we use an auxiliary variable method similar to (Teh et al., 2006) that we detail below. We sample θ_r based on three parts: 1) actual counts n_r associated with node r, 2) pseudo counts $\tilde{n}_r$, propagated from all children nodes of r and 3) topic proportion $\theta_{\pi(r)}$ from the parent node of r. Thus, topic proportions for node r are influenced by its children nodes and its parent node, enforcing topic proportion cascading on the tree.

To sample $\tilde{n}_r$, we start from all children node of r. Let $\tilde{s}_{p,k}$ be the number of counts that node $p \in C(r)$ will propagate to its parent node r and $n_{p,k}$ is the actual number of times topic k appears at node p. We sample $\tilde{s}_{p,k}$ by the following procedure. We firstly set it to 0, then for $j = 1, \cdots, n_{p,k} + \tilde{n}_{p,k}$, flip a coin with bias $\frac{\lambda\theta_{r,k}}{j-1+\lambda\theta_{r,k}}$, and increment $\tilde{s}_{p,k}$ if the coin turns head. The final value of $\tilde{s}_{p,k}$ is a sample from the Antoniak distribution. Thus, for node r, $\tilde{n}_{r,k} = \sum_{p \in C(r)} \tilde{s}_{p,k}$. This sampling procedure is done from the bottom to the top. Note that $\tilde{s}_{p,k}$ has the meaning as the number of times the parent node was visited when sampling topic k at node p.

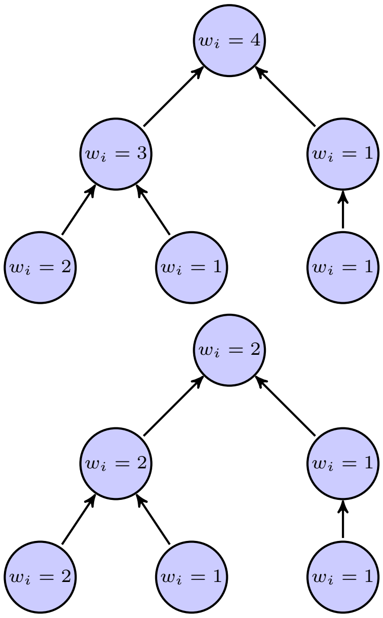

Figure 6. This is a demonstration of "Maximal Paths" (top) and "Minimal Paths" (bottom), showing how counts on leaf nodes propagate to the top. w_i is the number of times term w_i appearing on the node.

After smoothing over the tree from bottom to the top, we will have pseudo counts on each node. Thus, new topic proportions for each node can be effectively sampled by:

$$\theta_r \sim \text{Dir}\left(n_r + \tilde{n}_r + \lambda\theta_{\pi(r)}\right) \qquad (7)$$

where n_r is the actual count vector for node r and $\tilde{n}_r$ is the pseudo count vector. We do this process from the top to the bottom of the tree.

A.2 Sampling Regional Language Models

As we discussed before, regional language models are cascaded through the tree structure. Thus, we need to sample them explicitly in the inference algorithm. The sampling process is also a top-down procedure where we start from the root node. For the root node, we always sample it from a uniform Dirichlet distribution

$\phi_{\text{root}} \sim \text{Dir}(0.1/V, \cdots, 0.1/V)$. For all other nodes, we sample ϕ_r from:

$$\phi_r \sim \text{Dir}\left(m_r + \tilde{m}_r + \omega\phi_{\pi(r)}\right) \tag{8}$$

where m_r is the count vector for node r, $\tilde{m}_r$ is a smoothed count vector for node r and ω is a parameter. Here, $m_{(r,v)}$ is the number of times term v appearing in node r. For $\tilde{m}_r$, it is a smoothed vector of counts from sub-trees of node r. It can be sampled through a draw from the corresponding Antoniak distribution, similar to Section (8). However, since the element in ϕ_r is much larger than topic proportions, it is not efficient. Here, we adopt two approximations (Cowans, 2006; Wallach, 2008):

1. **Minimal Paths**: In this case each node $p \in C(r)$ pushed a value of 1 to its parent, if $m_{p,v} > 0$.
2. **Maximal Paths**: Each node r propagate its full count $m_{p,v}$ vector to its parent node.

The sum of the values propagated from all $p \in C(r)$ to r defines $\tilde{m}_r$. Although the sampling process defined here is reasonable in theory, it might be extremely inefficient to store ϕ values for all nodes. Considering a modest vocabulary of 100k distinct terms, it is difficult to keep a vector for each region. To address this we use the sparsity of regional language models and adopt a space efficient way to store these vectors.

A.4 Tree Structure Kalman Filter

For all latent regions, we sample their mean vectors as a block using the multi-scale Kalman filter algorithm (Chou et al., 1994). The algorithm proceeds in two stages: upward filtering phase and downward-smoothing phase over the tree. Once the smoothed posterior probability of each node is computed, we sample its mean from this posterior.

We define the following two quantities, Ψ_n to be the prior covariance of node n, i.e. the sum of the covariances along the path form the root to node n, and $F_n = \Psi_{\text{level}(n)-1}[\Psi_{\text{level}(n)}]^{-1}$, which are used to ease the computations below.

We first begin the upward filtering phase by computing the conditional posterior for a given node n based on each of its children $m \in C(n)$. Recall that each child 0 of every node specify the set of documents sampled directly from this node. Thus we have two different update equations as follows:

$$\begin{aligned} \Sigma_{n,0} &= \Psi_n \Sigma_{\pi(n)} \Big[\Sigma_{\pi(n)} + |C(n)|\Psi_n\Big]^{-1} \\ \mu_{n,0} &= \Sigma_{n,0}\Sigma_{\pi(n)}^{-1}\Big[\sum_{d \in C(n,0)} I_d\Big] \end{aligned} \tag{9}$$

$$\begin{aligned} \mu_{n,m} &= F_m \hat{\mu}_m \\ \Sigma_{n,m} &= F_m \Sigma_m F_m^T + F_m \Sigma_n \end{aligned} \tag{10}$$

where $m \in C(n)$. Once these quantities are calculated for all children nodes for n, we update the filtered mean and covariance of node n, $(\hat{\mu}_n, \hat{\Sigma}_n)$ based on its downward tree as follows:

$$\begin{aligned} \hat{\Sigma}_n &= \Big[\Psi_n^{-1} + \sum_{m \in C(n)} [\Sigma_{n,m}^{-1} - \Psi_n^{-1}]\Big]^{-1} \\ \hat{\mu}_n &= \hat{\Sigma}_n \Big[\sum_{m \in C(n)} \Sigma_{n,m}^{-1}\mu_{n,m}\Big] \end{aligned} \tag{11}$$

Once we reach the root node, we start the second downward smoothing phase and compute the smoothed posterior for each node (μ'_n, Σ'_n), as follows:

$$\mu'_{\text{root}} = \hat{\mu}_{\text{root}} \quad \Sigma'_{\text{root}} = \hat{\Sigma}_{\text{root}} \tag{12}$$

$$\begin{aligned} \mu'_n &= \hat{\mu}_n + J_n\Big[\mu'_{\pi(n)} - \mu_{\pi(n),n}\Big] \\ \Sigma'_n &= \Sigma_n + J_n\Big[\Sigma'_{\pi(n)} - \Sigma_{\pi(n),n}\Big] J_n^T \end{aligned} \tag{13}$$

where $J_n = \hat{\Sigma}_n F_n^T \hat{\Sigma}_{\pi(n)}^{-1}$. Here, $\Sigma_{\cdot,\cdot}$ and $\mu_{\cdot,\cdot}$ are from upward phase. After upward and downward updates, we sample the mean μ_n of each node n from $\mathcal{N}(\mu'_n, \Sigma'_n)$.

A.3 Sampling Topic Assignments

Given the current region assignment, we need to sample the topic allocation variable $z_{(d,i)}$ for word $w_{(d,i)}$ in document d:

$$\begin{aligned} &P(z_{(d,i)} = k \mid w, z_{-(d,i)}, r, l, \Theta, \Phi) \propto \\ &P(z_{(d,i)} = k \mid z_{-(d,i)}, r, \Theta, \Phi) P(w_{(d,i)} \mid z, w_{-(d,i)}, \Phi) \end{aligned}$$

Since all θ are integrated out, this is essentially similar to the Gibbs sampling in LDA where document-level topic proportions in LDA becomes region-level topic proportions. Thus, we can utilize a similar equation to sample topic assignments. Note, as we discussed in the last section, we have a $(T+1)$ matrix Π where the first dimension is a special row for regional language models that are distinct for each region. The sampling rule is as follows:

$$\begin{cases} \left(\tilde{n}_{r,k}^{-i} + n_{r,k}^{-i} + \rho\theta_{\pi(r),k}\right)\left[\frac{m_{k,v}^{-i} + \eta}{\sum_w m_{k,w}^{-i} + V\eta}\right] & k \neq 0 \\ \left(\tilde{n}_{r,0}^{-i} + n_{r,0}^{-i} + \rho\theta_{\pi(r),0}\right)\left[\frac{m_{r,v}^{-i} + \tilde{m}_{r,w} + \lambda\phi_{\pi(r),v}}{\sum_w m_{r,w}^{-i} + \tilde{m}_{r,w} + \lambda}\right] & k = 0 \end{cases} \tag{14}$$

where $v \equiv w_{(d,i)}$, $n_{r,k}$ is the number of times topic k appearing in region r and $m_{k,v}$ is the number of times term v assigned to k. Here, $n_{r,0}$ and $m_{r,v}$ serve the purpose for the special index for the regional language model. Note, n_*^{-i} and m_*^{-i} mean that the count should exclude the current token.

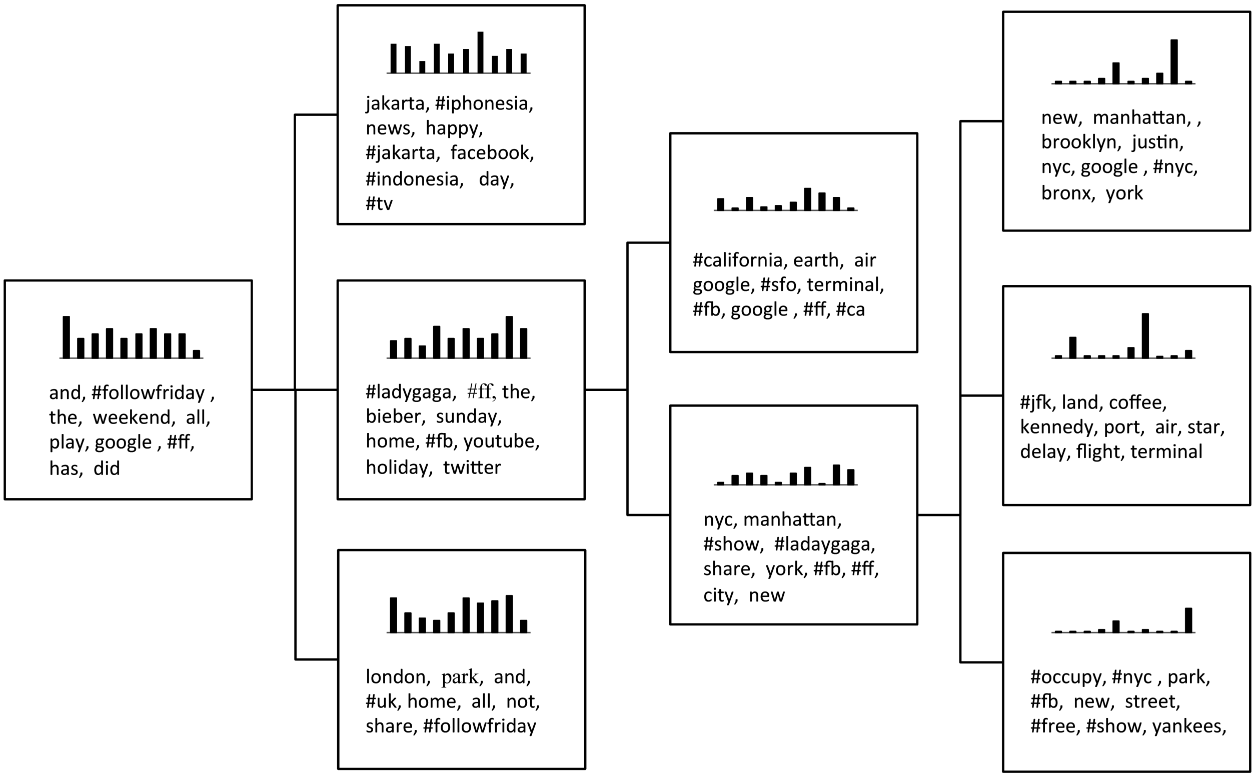

Figure 7. A small portion of the tree structure discovered from DS1.

APPENDIX B: Detailed Analysis of the Twitter dataset

B.1 User Location modeling

We demonstrate the efficacy of our model on two datasets obtained from Twitter streams. Each tweet contains a real-valued latitude and longitude vector. We remove all non-English tweets and randomly sample 10,000 Twitter users from a larger dataset, with their full set of tweets between January 2011 and May 2011, resulting 573,203 distinct tweets. The size of dataset is significantly larger than the ones used in some similar studies (e.g, (Eisenstein et al., 2010; Yin et al., 2011)). We denote this dataset as DS1. For this dataset, we split the users (with all her tweets) into **disjoint** training and test subsets such that users in the training set **do not** appear in the test set. In other words, users in the test set are like *new* users. This is the most adversarial setting. In order to compare with other location prediction methods, we also apply our model a dataset available at http://www.ark.cs.cmu.edu/GeoText, denoted as DS2, using the same split as in (Eisenstein et al., 2010). The priors over topics and topics mixing vectors were set to .1 and ω, λ to .1 favouring sparser representation at lower levels. The remaining hyperparameters are tunded using cross-validation. We ran the model until the training likelihood asymptotes.

Figure 7 provides a small *subtree* of the hierarchy discovered on DS1 with the number of topics fixed to 10. Each box represents a region where the root node is the leftmost node. The bar charts demonstrate overall topic proportions. The words attached to each box are the top ranked terms in regional language models (they are all in English since we removed all other content). Because of cascading patterns defined in the model, it is clear that topic proportions become increasingly sparse as the level of nodes increases. This is desirable as we can see that nodes in higher level represent broader regions. The first level roughly corresponds to Indonesia, the USA and the UK, under USA, the model discovers CA and NYC and then under NYC it discovers attraction regions. We show some global topics in Table 1 as well which are more generic than the regional language models.

B.2 Location Prediction

As discussed in Section 1, users' mobility patterns can be inferred from content. We test the accuracy by estimating locations for Tweets. Differing from (Eisenstein et al., 2010) who aim to estimate a *single* location

Table 5. Top ranked terms for some global topics.

Entertainment
video gaga tonight album music playing artist video
itunes apple produced bieber #bieber lol new songs
Sports
winner yankees kobe nba austin weekend giants
horse #nba college victory win
Politics
tsunami election #egypt middle eu japan egypt
tunisia obama afghanistan russian
Technology
iphone wifi apple google ipad mobile app online
flash android apps phone data

Table 6. Location accuracy on DS1 and DS2.

Results on DS1	**Avg. Error**	**Regions**
(Yin et al., 2011)	150.06	400
(Hong et al., 2012)	118.96	1000
Approx.	91.47	2254
MH	90.83	2196
Exact	83.72	2051

Results on DS1	**Avg. Error**	**Regions**
(Eisenstein et al., 2010)	494	-
(Wing & Baldridge, 2011)	479	-
(Eisenstein et al., 2011)	501	-
(Hong et al., 2012)	373	100
Approx.	298	836
MH	299	814
Exact	275	823

Table 7. Accuracy of different approximations and sampling methods for computing ϕ_r.

Method	DS1	DS2
Minimal Paths	91.47	298.15
Maximal Paths	90.39	295.72
Antoniak	88.56	291.14

Table 8. Ablation study of our model

Results on DS1	**Avg. Error**	**Regions**
(Hong et al., 2012)	118.96	1000
No Hierarchy	122.43	1377
No Regional Language Models	109.55	2186
No Personalization	98.19	2034
Full Model.	91.47	2254

Results on DS2	**Avg. Error**	**Regions**
(Hong et al., 2012)	372.99	100
No Hierarchy	404.26	116
No Regional Language Models	345.18	798
No Personalization	310.35	770
Full Model.	298.15	836

for each user (note that they use the location of the first tweet as a reference, which may not be ideal), our goal is to infer the location of each new tweet, based on its content and the author's other tweets.

Based on our statistics, only $1\% \sim 2\%$ of tweets have either geographical locations (including Twitter Places) explicitly attached, meaning that we cannot easily locate a majority of tweets. However, geographical locations can be used to predict users' behaviors and uncover users' interests (Cho et al., 2011; Cheng et al., 2011) and therefore it is potentially invaluable for many perspectives, such as behavioral targeting and online advertisements. For each new tweet (from a new user not seen during training), we predict its location as $\hat{l}_d$. We calculate the Euclidean distance between predicted value and the true location and average them over the whole test set $\frac{1}{N}\sum l(\hat{l}_d, l_d)$ where $l(a, b)$ is the distance and N is the total number of tweets in the test set. The average error is calculated in kilometres. We use three inference algorithms for our model here: 1) exact algorithm denoted as Exact, 2) M-H sampling, denoted as MH and 3) the approximation algorithm as Approx..

For DS1 we compare our model with the following approaches:

Yin 2011 (Yin et al., 2011) Their method is essentially to have a global set of topics shared across all latent regions. There is no regional language models in the model. Besides, no user level preferences are learned in the model.

Hong 2012 (Hong et al., 2012) Their method utilizes a sparse additive generative model to incorporate a background language models, regional language models and global topics. The model also considers users' preferences over topics and regions as well.

For all these models, the prediction is done by two steps: 1) choosing the region index that can maximize the test tweet likelihood, and 2) use the mean location of the region as the predicted location. For Yin 2011 and Hong 2012, the regions are the optimal region which achieves the best performance. For our method, the regions are calculated as the average of number of regions from several iterations after the inference algorithm converges. The results are shown in the top part of Table 6.

The first observation is that all three inference algorithms outperforms Yin 2011 and Hong 2012 significantly. Note that for both Yin 2011 and Hong 2012, we need to manually tune the number of regions as well as the number of topics, which requires a significant amount of computational efforts, while for our model, the number of regions grows naturally with the data. Also, we notice that the number of regions for the optimal performed model inferred by all three inference algorithms is larger than its counterparts Yin 2011 and Hong 2012. We conjecture that this is due to the fact that the model organizes regions in a tree-like structure and therefore more regions are needed to represent the fine scale of locations. In addition, we observe that `Exact` indeed performs better than `Approx.` and `MH`.

For the comparison on the `DS2` dataset, we compare with:

(Eisenstein et al., 2010) The model is to learn a base topic matrix that can be shared across all latent regions and a different topic matrix as the regional variation for each latent region. No user level preferences are learned in the model. The best reported results are used in the experiments.

(Eisenstein et al., 2011) The original `SAGE` paper. The best reported results are used in the experiments.

(Wing & Baldridge, 2011) Their method is essentially to learn regional language models per explicit regions.

(Hong et al., 2012) This was the previous state of the art.

For (Eisenstein et al., 2010; Wing & Baldridge, 2011; Eisenstein et al., 2011), the authors do not report optimal regions. For (Hong et al., 2012), the optimal region is reported from the paper. The best reported results are used in the experiments. For our method, the regions are calculated as the same fashion as above. The results are shown in the second part of Figure 6. It is obvious that our full model performs the best on this public dataset. Indeed, we have approximately 40% improvement over the best known algorithm (Hong et al., 2012) (note that area accuracy is quadratic in the distance). Recall that all prior methods used a flat clustering approach to locations. Thus, it is possible that the hierarchical structure learned from the data helps the model to perform better on the prediction task.

In Section 8, we discussed how regional language models can be sampled. Here, we compare the two approximation methods and directly sampling from Antoniak distributions based on `Approx.`, shown in Table 7. We can see that all three methods achieve comparable results although sampling Antoniak distributions can have slightly better predictive results. However, it takes substantially more time to draw from the Antoniak distribution, compared to Minimal Paths and Maximal Paths. In Table 6, we only report the results by using Minimal Paths.

B.3 Ablation Study

In this section, we investigate the effectiveness of different components of the model and reveal which parts really help with the performance, in terms of location prediction. For both `DS1` and `DS2`, we compare the following versions:

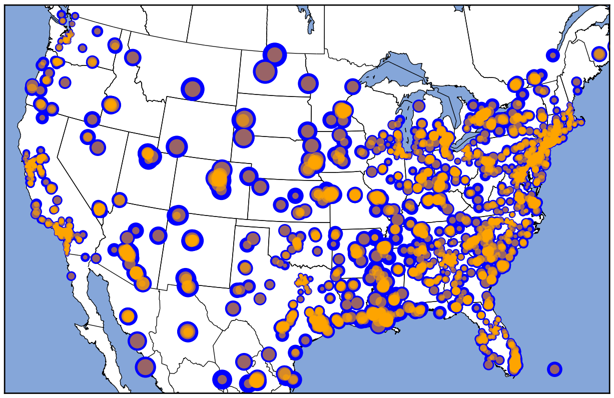

Figure 8. Error analysis for the state-of-the-art model (Hong et al., 2012) (blue circles) and our model (orange circles) on `DS1`.

`No Hierarchy` In this model, we do not have a hierarchical structure of regions while the number of regions is still infinite. Regional language models and a set of global topics are utilized.

`No Regional Language Model` No regional language model version of our proposed model: In this model, we still have the hierarchical structure over regions but no only having a global set of topics without regional language models.

`No Personalization` No personal distribution over the tree structure: In this model, we assume that all tweets are generated by a fictitious user and essentially no personal preferences are incorporated.

`Full Model` Our full model using the approximation sampling algorithm.

The results are shown in Table 8. The first observation is that all variants which utilize hierarchical structures of regions are better than other methods. This validates our assumption that hierarchies of regions can control the scope of regions and therefore smaller regions can be discovered from the data. This is also clearly observable from the optimal number of regions these methods have discovered. For `No Regional language Model`, it is only slightly better than `Hong` as it does not incorporate regional language models into account. We can see the effect of regional language models by focusing on `No Personalization` where no personal distributions over the tree is introduced. In summary, `Full Model.` demonstrated that personalized tree structures can further boost the performance.

B.5 Error Analysis

In order to understand how our model performs in terms of prediction we conduct a qualitative error analysis on our model as well on the the state-of-the-art model (Hong et al., 2012) on all users in the USA on `DS1`. The results are given in Figure 8. Each circle in the map represents 1000 tweets. The magnitude of the circle represents the magnitude of **average** error made for these 1000 tweets. Note that the circles are re-scaled such as to be visible on the map (i.e. radii do not correspond to absolute location error).

We observe that in the industrialized coastal regions both models perform significantly better than in the Midwest. This is because that we have more users in those areas and therefore we can, in general, learn better distributions over those regions. At the same time, users in those areas might have much more discriminative mobility patterns relative to users in the Midwest. The second observation is our method consistently outperforms (Hong et al., 2012). This is particularly salient in the Midwest.