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

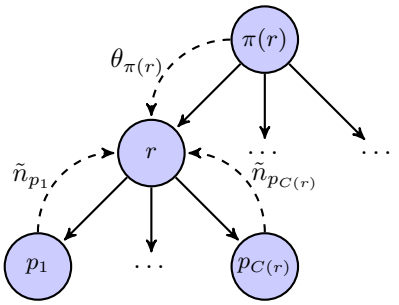

Figure 5. This is a demonstration of sampling θ_r, the distribution over topics for node r. The sampling is drawn from a Dirichlet distribution with parameters consisting of count statistics n_r from node r, pseudo counts $\tilde{n}_r$ gathering from its children nodes and topic proportions $\theta_{\pi(r)}$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

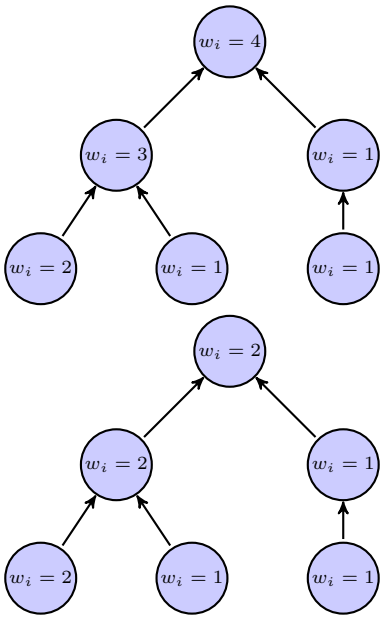

Figure 6. This is a demonstration of "Maximal Paths" (top) and "Minimal Paths" (bottom), showing how counts on leaf nodes propagate to the top. w_i is the number of times term w_i appearing on the node.

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:

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

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:

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

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

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:

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

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 $J_n = \hat{\Sigma}_n F_n^T \hat{\Sigma}^{-1}_{\pi(n)}$. Here, $\Sigma_{.,.}$ and $\mu_{.,.}$ 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:

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

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:

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

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 hyper-parameters 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

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.

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 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

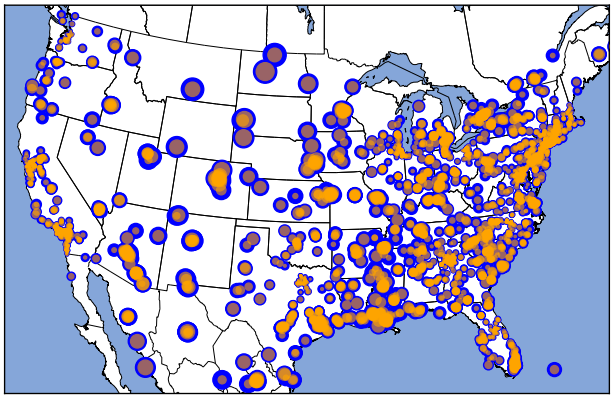

Figure 8. Error analysis for the state-of-the-art model (Hong et al., 2012) (blue circles) and our model (orange circles) on `DS1`.

really help with the performance, in terms of location prediction. For both `DS1` and `DS2`, we compare the following versions:

`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.