Hierarchical Geographical Modeling of User Locations from Social Media Posts*

Amr Ahmed[§], Liangjie Hong[†], Alex Smola[§,⋄]
§ Google, Mountain View, CA 94043, USA
† Yahoo! Labs, 701 First Ave., Sunnyvale, CA 94089, USA
⋄ Carnegie Mellon University, Pittsburgh, PA 15213, USA
amra@google.com, lih307@cse.lehigh.edu, alex@smola.org

ABSTRACT

With the availability of cheap location sensors, geotagging of messages in online social networks is proliferating. For instance, Twitter, Facebook, Foursquare, and Google+ provide these services both explicitly by letting users choose their location or implicitly via a sensor. This paper presents an integrated *generative* model of location and message content. That is, we provide a model for combining distributions over locations, topics, and over user characteristics, both in terms of location and in terms of their content preferences. Unlike previous work which modeled data in a flat pre-defined representation, our model automatically infers both the hierarchical structure over content and over the size and position of geographical locations. This affords significantly higher accuracy — location uncertainty is reduced by 40% relative to the best previous results [21] achieved on location estimation from Tweets.

We achieve this goal by proposing a new statistical model, the nested Chinese Restaurant Franchise (nCRF), a hierarchical model of tree distributions. Much statistical structure is shared between users. That said, each user has his own distribution over interests and places. The use of the nCRF allows us to capture the following effects: (1) We provide a topic model for Tweets; (2) We obtain location specific topics; (3) We infer a latent distribution of locations; (4) We provide a joint hierarchical model of topics and locations; (5) We infer personalized preferences over topics and locations within the above model. In doing so, we are both able to obtain accurate estimates of the location of a user based on his tweets and to obtain a detailed estimate of a geographical language model.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; [I.2.6] [**Artificial Intelligence**]: Learning – Parameter Learning; I.2.7 [**Artificial Intelligence**]: Natural Language Processing

Keywords

Geolocation; Twitter; Topic Models; User Profiling; Nonparemetric Bayesian Models; Chinese Restaurant Process

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. Moreover, they are widely used for disseminating information in emergencies, such as the Tsunami in Japan or Hurricane Sandy in New York. Moreover, Twitter was used extensively in elections and played an instrumental role in facilitating the Arab Spring.

In addition to its use as a content sharing platform, micro-blogging services like Twitter, along with other location sharing services such as Foursquare, Gowalla, Facebook Places or Google+ are nowadays supporting location services. That is, users are able to specify their location in messages, either explicitly, by letting users choose their place, or implicitly, by enabling geo-tagging functionality. Such pervasive geotagging is facilitated by the availability of cheap GPS sensors and improved location based services, e.g. via the use of WiFi fingerprinting in mobile devices.

1.1 Related Work

This wealth of data presents an exciting opportunity for statistical modeling. For instance we may analyze the relationship between content creation and sharing and conversely, this allows us to infer geographic location purely based on user-generated content. These issues have attracted significant attention [23, 28, 17, 13, 12], however, incorporating geographical information, language models and user preferences simultaneously is nontrivial. Many models only cover some aspects of the problem described above while ignoring the remainder. For instance, [30] partition the earth into equally sized grids and learn language models per area. This approach has two important shortcomings: the size or the number of grids is pre-defined and cannot be adapted efficiently according to the underlying distribution. Secondly, the model loses the ablity to uncover global topics shared across areas. On the other hand, [24] use uneven grids. However their work ignores shared interests among users. Moreover, the hidden structure inherent in the distribution over regions is ignored.

Another line of research [17, 16, 21] takes regional language variations and global topics into account by bridging *finite* mixture Gaussian models and topic models. Unlike pre-defined grids or regions, these models usually employ a flat clustering model of locations. This flat structure is unnatural in terms of the language model: while it is reasonable to assume that New Yorkers and San Franciscans might differ in terms of the content of the tweets, it is also

[*] This work was performed while AA and AS were with Yahoo! Research.

WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2035-1/13/05.

reasonable to assume that as a whole American tweets are more similar to each other, than to tweets from Egypt, China or Germany. As a side effect, location prediction is not always satisfactory. For instance the languge model of some big regions can degenerate to stopwords [17] while smaller regions might correspond to airports [21].

1.2 Key Contributions

In this paper we describe a *joint hierarchical* model of location and content, both personalized to the individual preferences of a user. That is, we build a generative hierarchical model for topics occurring in tweets. That is, we assume that topics occurring at lower levels of the tree are more specific versions of the topics found closer to the root level. We perform a similar approach when dealing with global topics. There their *distribution* is assumed to follow a corresponding hierarchical structure. Moreover, we assume that these topics are location specific, that is we adorn this tree with locations at which such topics could be observed. Finally, we represent users by modeling distributions over leaves within the overall tree. That is, we encode location specific generative models for each user.

Bayesian nonparametrics is rich in structured and hierarchical models. Nonetheless, we found that no previously proposed structure was a good fit for our needs, hence the need the model proposed in this paper — the nested Chinese Restaurant Franchise (nCRF). It addresses the following key problem: 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.

Our model allows us to discover a hidden tree structure with unbounded width and depth while allowing users to have different distributions over this structure. The number of regions and their hierarchical structure are learned from data, and language models as well as topic distributions are cascaded over the tree structure, resulting in more natural clustering of tweets and locations. We develop an efficient algorithm to perform posterior inference and apply the framework to organize tweets into a hierarchical structure and show that this tree structure can be used to better predict locations of unlabeled tweets, resulting in significant improvements to state-of-the-art approaches. In addition, we show some interesting hidden patterns that are revealed through this hierarchical modeling. These experiments provide theoretical validation of the model.

This model is experimentally validated by demonstrating significantly better accuracy (39% and 45% respectively on two benchmark datasets relative to the best prior work [21]) in inferring the location of a tweet without having access to its geographical location. Secondly, we are able to annotate locations with topics without suffering from any of the degeneracies inherent in [17, 21]. These topics form a human-understandable hierarchy, generated *automatically* from data without the need for manual intervention. This can be used, e.g. for improved content targeting and representation. It also provides improved information regarding the locations and topical preferences of individual users.

2. BACKGROUND

Given the considerable amount of prior work both on Bayesian Nonparametrics and on the analysis of user generated content it is worth while to discuss these issues in further detail before introducing the nested Chinese Restaurant Franchise and its application to Tweets. We begin with a summary of key aspects when modeling microblogs.

2.1 Modeling Microblogs

There is a substantial body of research on geographical language modeling. We review some key threads:

[23] propose a model based on Probabilistic Latent Semantic Indexing (PLSA) [20]. It assumes that each word is either drawn from a universal background topic or from a location and time dependent language model. However, the mixture coefficients between the background topic and other spatio-temporal topics ones is tuned manually. Since the model uses PLSA, no prior distribution is (or could be) assumed. Evaluation is performed via anecdotal results.

[28] introduce a fully Bayesian generative model. Rather than working with actual locations, they fixed a number of region labels and assume that each term is associated with a location label. For each word in a document, a topic assignment is first generated according to a multinomial distribution. Then the term and the location are generated dependent on this topic assignment. Again the evaluation is limited to anecdotal results.

[25] propose a similar model. For evaluation they measure Deviation Information Criteria (a model complexity criterion similar to BIC), as well as classification accuracy using manually labeled data. One of the drawbacks of the work is that they only use data from Flickr restricted to the greater London area. [19] extend [28] by introducing the notion of global and local topics. Inference uses Gibbs Sampling.

[17] propose a model utilizing the correlations between global and local topics. In their model, each author is assigned a latent region variable and an observed GPS location. Terms and the actual GPS location are both conditioned on the latent region variable. The topics to generate terms are local topics, which are derived from global topics. For inference [17] use Variational EM. For evaluation purposes the accuracy of predicted location is used.

[31] propose a model similar in spirit to [17]. The terms and the location of a particular document are generated by a latent region. The location is generated from a region by a normal distribution and the region is sampled from a multinomial distribution. However, inference is performed using MAP-style EM rather than a fully Bayesian approach. [30] use an even simpler approach where documents are assigned to geodesic grids and thus a supervised learning method is utilized, essentially via a naïve Bayes classifier on the grid.

Finally [13] studied human mobility in location sharing services. They found that users tend to appear in a very limited number of places (e.g., office and home). They demonstrated that it might be effective enough to use a two component Gaussian mixture model to estimate users' locations.

2.2 The Chinese Resturant Process

We now proceed to reviewing some key components of the statistical toolkit we employ for content modeling. One of the prototypical ingredients for nonparametric modeling is the Dirichlet Process $DP(H, \gamma)$ [18, 8, 10]. It allows for a discrete distribution of observations drawn from an arbitrary base measure H. We write $G_0 \sim DP(H, \gamma)$ to denote a draw from a DP where γ controls the variance of the draws around the base measure. G_0 is itself a distribution over an infinte number of components. We then draw parameters $\theta_i \sim G_0$. Placing this prior on top of a mixture model we get the Drichlet process mixture model (DPM). In DPM we extend the aforementioned generative process to finally draw the observed data points x_i from θ_i: $x_i \sim f(\theta_i)$.

A useful view of the Dirichlet Process is the *Chinese Restaurant metaphor*. In it each data point is considered as a customer in a restaurant with an infinite number of tables. Initially all tables are empty. Customers pick existing tables in proportion to their popularity. In it the probability for customer i to pick table j is

$$\Pr\{z_i = j\} = \begin{cases} \frac{N_j}{\sum_k N_k + \alpha} & \text{for an existing table} \\ \frac{\alpha}{\sum_k N_k + \alpha} & \text{for a new table.} \end{cases} \quad (1)$$

Here z_i encodes the choice of customer i. N_j denotes the *current* number of customers sitting at table j and $\sum_k N_k$ is the total number of customers so far. This makes the Chinese Restaurant Process a single parameter distribution over partitions of the integers. The 'dish' $x_j \sim H$ chosen at table j is drawn iid from H, the base measure. i

2.3 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 [26,]. In other words, rather than

$$G \sim DP(H, \gamma) \text{ we now have } G_i \sim DP(G_0, \gamma') \quad (2)$$

$$\text{and } G_0 \sim DP(H, \gamma) \quad (3)$$

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 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 [26, 9]. 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 at the Chinese Restaurant Franchise (CRF). In it each restaurant maintains its set of tables but shares the same set of mixtures. A customer at restaurant k can 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. In this global distribution, a dish (mixture) is chosen proportional to its use across restaurants, however, a new global dish can be chosen with probability proportional to γ.

2.4 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 [11]. 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 tool, it is still only allows each document to have a single path over the tree – a limitation that our model in Section 3 will remedy.

3. THE NESTED CHINESE RESTAURANT FRANCHISE PROCESS

We are now in a position to introduce the Nested Chinese Restaurant Franchise (nCRF). As its name suggests, it borrows both from the Chinese Restaurant Franchise, thus allowing us to share strength between groups, and the Nested Chinese Restaurant Process, thus allowing us to obtain a hierarchical distribution over observations. Although the Nested Chinese Restaurant Process, introduced in [11], provides a convenient way to impose a distribution over tree-like structures, it is difficult to apply it directly in our settings due to the reason that the nCRP-induced distribution over the hierarchy is a *global* distribution shared across all data partitions, such as documents. Instead, in our case, we wish to have a *personalized* distributions over the same hierarchy for each user. Subsequently, we adorn each vertex in the tree with a generative model to represent language and topic cascades. In the context of spatial modeling of user generated content, for instance, each node in the tree represents a geographical region. In the context of document modeling, we will associate each document with an nCRP and tie together documents using the franchise. Details of the nCRF, as applied to these problems are given in later sections. For now we focus on the generative process itself.

3.1 Basic Idea

Our goal is to design a non-parametric model over trees, where each user has its own tree, but the set of nodes in the trees, and their structure, such as parent-child relationships, are shared across all users. This process is illustrated in Figure 1. In a nutshell, we achieve this by associating an nCRP process with each user.

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 similarly. 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 process the nested CRF process. An equivalent representation is the nHDP process, where the base measure at nodes A_1

and A_2 is sampled from the base measure at node A. The base measure at node A is in turn sampled from a global base measure over the integers. Note that the global tree is just a superset of all user trees, and each user only places some probability mass in selected regions he visits. Once the trees are generated, we define a cascading process of node attributes over the tree.

3.2 A Chinese Restaurant Metaphor

Consider the case where we want to generate a path for a tweet written 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 (in this user's tree) 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). All selection probabilities are governed using the standard CRF's rich-gets-rich mechanism. Once a child node is selected, the process recurses with that node until a full path is defined.

We need some notation regarding the structure of the trees over different topics. We denote by i a vertex in the tree. level(i) denotes the level of node i, $C(i)$ denotes the children and $\pi(i)$ its parent. Moreover, n_{ij} is the number of times child j is selected at node i when generating data and $n_i := \sum_j n_{ij}$ is the number of (non-unique) children of i. Moreover, we use the superscript u to index the above quantities for each user. That is, n_i^u, n_{ij}^u, $C^u(i)$ are matching user-specific quantities. Clearly $\pi^u(i) = \pi(i)$ since the user trees are mapped to the common tree. Also note that usually $C^u(i) \subset C(i)$, i.e. we only select a subset of nodes in the user tree.

This allows us to specify the collapsed generative probabilities at vertex i. As per the Chinese Restaurant metaphor the probability of selecting an existing child node is

$$\Pr\{\text{child } j \text{ at } i\} = \begin{cases} \frac{n_{ij}^u}{n_i^u+\alpha} & \text{if } j \in C^u(i) \text{ is from the user tree} \\ \frac{\alpha}{n_i^u+\alpha} & \text{otherwise} \end{cases} \tag{4}$$

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:

$$\Pr\{\text{child } j \text{ at } i\} = \begin{cases} \frac{n_{ij}}{n_i+\beta} & \text{if } j \in C(i) \\ \frac{\alpha}{n_i+\beta} & \text{if this is a new child} \end{cases} \tag{5}$$

Combining (4) and (5) we have the full probability as

$$\Pr\{\text{child } j \text{ at } i\} = \begin{cases} \frac{n_{ij}^u}{n_i^u+\alpha} + \frac{\alpha}{n_i^u+\alpha}\frac{n_{ij}}{n_i+\beta} & \text{if } j \in C^u(i) \\ \frac{\alpha}{n_i^u+\alpha}\frac{n_{ij}}{n_i+\beta} & \text{if } j \in C(i) \backslash C^u(i) \\ \frac{\alpha}{n_i^u+\alpha}\frac{\beta}{n_i+\beta} & \text{if } j \notin C(i) \end{cases}$$

Note here that we used a direct-assignment representation for the CRF at each node to avoid overloading notations with maintaining different tables for the same child at each node. This is correct due to the coagulation / fragmentation equivalence in Dirichlet Processes [22]. 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. During inference, an axillary variable method is used to link the local n_{ij}^u and global counts n_{ij} using the Anotoniak distribution as described by [26].

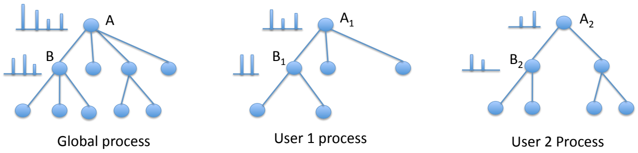

Figure 1: An illustration of the nested Chinese Restaurant Franchise involving a common tree over components (left) and two subtrees representing processes for two separate subgroups (e.g. users on Twitter). Each user samples from his own distribution over topics, smoothed by the global process. Thus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

Once a child node is selected, the process is repeated until a full path is defined. To ensure that the path terminates we need to add the notion of an 'exit child'. We use sequence notation to index a vertex in the tree (e.g. 2:5:3:1 means choosing a path which uses the second, fifth, third and then first vertex in the path respectively) we define child 1 to be the terminator. In other words, whenever we select child 1 anywhere on the path, sampling terminates. At this point the observation is generated using the global parameters associated with node i.

Two important notes are in order here: We keep a global index of the children under each node across all users and track which of them are materialized under each user's tree using the list $C^u(.)$. Hence giving a special semantic to child 1 under each node is valid. Second, we stress that indices are only unique under each parent node, that is, we need to specify the entire path when implementing the inference routines. However, when referring to a generic node in the tree we simply refer to node i, ignoring the fact that i is in fact, a slightly more complex data structure tracing the path from root to a particular vertex. Note that by construction this sequence must end in a 1, i.e. the exit child. Also note that we could use a different smoothing probability for the first vertex as was done in [1].

4. GENERATING MICROBLOGS

Microblogs are a rather unique form of expression. They are concise, often very situation and context specific, and they exhibit characteristics unique to to their authors. We want to capture this in a generative model. More to the point, 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. We use tweets and documents interchangeably to mean the same object.

A natural assumption is that locations come with their own location specific topics, such as airport names, local sports teams, politicians, companies, festivals, language idiosyncrasies, foreign languages, etc.; That said, it is reasonable to assume that some topics are globally popular. Nonetheless, their relative degree of popularity is likely to be location specific. One would assume that these preferences correlate hierarchically. That is, quite likely tweets in California are more similar to those in Oregon than, say, in

Bavaria. This suggests that it might be possible to arrange content and location preferences in a tree.

The above reasoning leads to a tree for arranging locations, local topics, and general topic preferences in a structured fashion. In statistical terms this means that we will model locations in a hierarchical tree-wise fashion. 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.

While these steps are useful in their own right, they do not yet resolve the issue of user specific preferences. That is, users commonly only frequent a small number of places. Moreover, they only tend to write about a relatively small subset of topics. Hence, a global tree-based hierarchical model is unlikely to be a good fit for the tweets generated by individuals. A better approach is to 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 a better content description. This leads to an improved model of both content and location. In fact, we find that our model is significantly beter than any previous location estimator using Tweets. We now map the above intuitive description of the process to a concrete model of data generation.

Tree distribution: This was discussed in the context of the nested Chinese Restaurant Franchise above. Recall that at each vertex i we assume that the distribution over children follows a CRP process.

The first component corresponds to the probability mass of observations remaining at the present vertex whereas all other components correspond to proper children of i. Note that this means that different vertices may have different numbers of children. It also means that with probability 1 any path in the tree terminates after a finite number of steps. This greatly simplifies inference since we never need to instantiate an infinite hierarchy.

Hierarchical location model: In analogy to [1] we consider a hierarchical multivariate Gaussian model. The main distinction is that we need not instantiate a shrinkage step towards the origin at each iteration (this would be ill defined on the surface of the earth). Instead, we simply assume an additive Gaussian model. We are able to achieve this simply since we assume decreasing variance when traversing the hierarchy, whereas [1] did not impose such a constraint. This yields the following model:

$$\mu_r \sim \mathcal{N}\left(\mu_{\pi(r)}, \Sigma_{\pi(r)}\right) \text{ and } \Sigma_r = \frac{1}{\mathsf{level}(r)}\Sigma_0. \qquad (6)$$

Here Σ_0 is the covariance matrix of the root node, and μ_r, Σ_r are the mean vector and covariance matrix of region r. and 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.

Generic topics: In our model we assume that we have a finite number of topics T (that this could easily be alleviated but we found that this was unnecessary in our experiments and it made it more difficult to predict well bounded memory footprint). We denote each global topic by Π_i. It is drawn from a Dirichlet distribution over V words:

$$\Pi_i \sim \mathrm{Dir}(\eta). \qquad (7)$$

Note that one could add hierarchical language models over topics (or longer range n-grams as discussed e.g. by [29]). This is likely to improve estimation quality for longer and structured texts.

Location specific language model: 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

$$\phi_0 \sim \mathrm{Dir}(\eta). \qquad (8)$$

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

$$\phi_r \sim \mathrm{Dir}\left(\omega\phi_{\pi(r)}\right) \qquad (9)$$

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: A similar construction can be used for hierarchically modeling *distributions* over topics hierarchically. 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

$$\theta_0 \sim \mathrm{Dir}(\beta) \qquad (10)$$

$$\theta_r \sim \mathrm{Dir}\left(\lambda\theta_{\pi(r)}\right) \qquad (11)$$

User specific tree distribution: Again, this is as described in Section 3. We use the distribution over the common tree as a prior and draw a user specific tree distribution over regions and associated topics.

In summary, the generative process is as follows: for each tweet d by user u, we firstly use nCRF to choose a node (latent region) r. Once this node is chosen, both the content of the document (words) and the geographical location, i.e. latitude and longitude, are generated from corresponding distributions of this node. For the geographical location, the generative process is straightforward as it is drawn from the regional dependent multivariate normal distribution. For each word w in the document d, we firstly choose a topic assignment z from the regional dependent topic distribution. Then, a word w is generated from the corresponding language model.

We have two sets of language models: global topics encoded in a global matrix Π and regional language models ϕ_i, one per latent region. Following the definition of the model one would need to introduce a Bernoulli random variable to determine which distribution might be the source for the word. However, this will introduce unnecessary overhead for bookkeeping and sampling. Instead, we use a compact representation exploiting the fact that a the mixture between multinomials is multinomial. The idea is to augment the global topic matrix with an additional topic (i.e. topic $T+1$) and reserve this topic for the the regional language model based on i_d. Every time this special index is sampled from the topic distribution, we just refer to the Tweet's regional language model. More specifically, we use $\bar{\Pi}$ to denote a $(T+1)$-row matrix, which combines the regional language model and the original global topic matrix. Using this trick we can eliminate the Bernoulli switch variable. This yields the following generative process:

1. For each tweet d written by each user u:
   (a) Sample a node $r_d \sim \text{nCRF}(\gamma, \alpha, u)$.
   (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. $\phi_{r_d} \sim \text{Dir}\left(\omega\phi_{\pi(r_d)}\right)$
      iii. $\theta_{r_d} \sim \text{Dir}\left(\lambda\theta_{\pi(r_d)}\right)$
   (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 $z_{(d,i)} \sim \text{Multi}(\theta_{r_d})$.
      ii. Sample word $w_{(d,i)} \sim \text{Multi}(\bar{\Pi}_{z_{(d,i)}})$.

We repeat this generative process for all users in the corpus.

5. INFERENCE ALGORITHM

In this section, we describe a collapsed direct-assignment Gibbs sampling algorithm to infer the latent structures imposed by the generative process introduced above. The observed variables needed by the Gibbs inference algorithm are: $w_{(d,i)}$, the i-th word in document d and l_d, the geographical location of document d. In addition to these input data, we also observe the author, as an index, of each documents. We collapse all multinomial variables and alternate sampling a region assignment r_d for each document/tweet d from each *user*, a topic assignment $z_{(d,i)}$ for each word in the document. Furthermore, we need to sample the parameters (a mean vector and a covariance matrix) for multivariate normal distributions to be able to calculate the probabilities for geographical locations, and the mean-geographical parameters for latent regions. We employ an auxiliary variable method similar to [26] to deal with the cascading behavior of topic mixing vectors θ and ϕ. Each of the following Sections detail each step.

5.1 Sampling Region Assignments

Instead of sampling a path r_d, as a block as in the nCRP [11], we use a level-wise strategy to sample a latent region for each tweet[1]. Where r_d is a specification of the path from the root to the selected latent region. Lets say starting from the root node, a document d with author u reached node i on the tree, then it can descend the tree further as follows:

1. stay on the current node i – i.e. pick child 0, and set $r_{d,i} = 0$.
2. Move to a child node j of i other than child 0, and set $r_{d,i} = j$
3. create a new child node of node i and move to it, and set $r_{d,i}$ accordingly.

Algorithm 1: The sketch of one iteration of the inference algorithm. D is the total number of documents

```
Sampling region and topic assignments:
for $d = 1$ to $D$ do
    Sample region assignments $r_d$, Eqn. (12)
    Sample topic assignments $z_d$, Eqn. (15)
Tree structure Kalman filter:
    From the bottom to top:
        Perform Eqn. (16, 17, 18)
    From the top to the bottom:
        Perform Eqn. (19,20)
Sampling topic proportions:
    From the bottom to the top:
        Sample $\tilde{n}_r$ from $C(r)\forall r$
    From the top to the bottom:
        Sample $\theta_r$, Eqn. (13)
Sampling regional language models:
    From the bottom to the top:
        Compute $\tilde{m}_r$ from $C(r)\forall r$
    From the top to the bottom:
        Sample $\phi_r$, Eqn. (14)
```

The probability of each choice shares a similar form:

$$P(r_{d,i} = \text{node} \,|\, r_{-d}, \text{rest})P(w_d, l_d, z_d \,|\, r, \text{rest}) \qquad (12)$$

where the second term in the right hand side is essentially the probability of the data given a choice of the node, which can be decomposed into three componenets: the probability of the document (terms) $P(w_d \,|\, r_d, \text{rest})$, the probability of the location $P(l_d \,|\, r_d, \text{rest})$, and the probability of the topic indicators $P(z_d \,|\, r_d, \text{rest})$. For $P(l_d \,|\, r, \text{rest})$, it is indeed $P(l_d \,|\, \mu_{r_d}, \Sigma_{r_d})$, evaluating the multivariate normal distribution associated to the corresponding node at l_d (This is because we explicitly represent these normal parameters, which is discussed in Section (5.5)). The component corresponding to $P(w_d \,|\, r, \text{rest})$ and $P(z_d \,|\, r, \text{rest})$ are just standard Dirichlet-multinomial integrals which reduces to the ration of two log-partition functions. Finally, the component $P(r_d = \text{node} \,|\, r_{-d}, \text{rest})$, is computed according to the nCRF process defined in Section (3.2).

Once a decision is made over node i and a child (other than 0) is selected, we descend with that child and repeat the process. The key idea here is that each node among the children of i acts as proxy for the subtree rooted under it due to the cascading nature of how the parameters are defined, and as such allows us to make an informed decision at each node separately without sampling the path as a block.

5.2 Sampling Topic Proportions

Since topic proportions for different regions are linked through the cascading process defined in Equation (11), we use an auxiliary variable method similar to [26] 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

[1] This approximate strategy gives close results to the exact method that samples a path as a block, while being much faster. A detailed comparison is left to the full version of this paper.

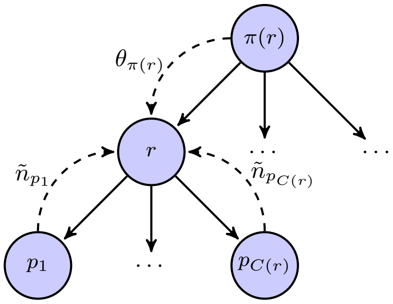

Figure 2: This is a demonstration of sampling θ_r, the distribution over topics for node r. The sampling is drawn from a Dirichlet distribution with parameters consisting of count statistics n_r from node r, pseudo counts $\tilde{n}_r$ gathering from its children nodes and topic proportions $\theta_{\pi(r)}$ from its parent node.

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.

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:

$$\theta_r \sim \text{Dir}\left(n_r + \tilde{n}_r + \lambda\theta_{\pi(r)}\right) \tag{13}$$

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.

5.3 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:

$$\phi_r \sim \text{Dir}\left(m_r + \tilde{m}_r + \omega\phi_{\pi(r)}\right) \tag{14}$$

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 (5.2). However, since the element in ϕ_r is much larger than topic proportions, it is not efficient. Here, we adopt two approximations [15, 27]:

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.

5.4 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:

$$\begin{cases} \left(\tilde{n}_{r,k}^{-i} + n_{r,k}^{-i} + \rho\theta_{\pi(r),k}\right)\left[\frac{m_{k,v}^{-i}+\eta}{\sum_w m_{k,w}^{-i}+V\eta}\right] & k \neq 0 \\ \left(\tilde{n}_{r,0}^{-i} + n_{r,0}^{-i} + \rho\theta_{\pi(r),0}\right)\left[\frac{m_{r,v}^{-i}+\tilde{m}_{r,w}+\lambda\phi_{\pi(r),v}}{\sum_w m_{r,w}^{-i}+\tilde{m}_{r,w}+\lambda}\right] & k = 0 \end{cases} \tag{15}$$

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.

5.5 Tree Structure Kalman Filter

For all latent regions, we sample their mean vectors as a block using the multi-scale Kalman filter algorithm [14]. 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:

$$\begin{aligned} \Sigma_{n,0} &= \Psi_n \Sigma_{\pi(n)} \Big[\Sigma_{\pi(n)} + |C(n)|\Psi_n\Big]^{-1} \\ \mu_{n,0} &= \Sigma_{n,0}\Sigma_{\pi(n)}^{-1}\Big[\sum_{d \in C(n,0)} I_d\Big] \end{aligned} \tag{16}$$

$$\begin{aligned}\mu_{n,m} &= F_m \hat{\mu}_m \\ \Sigma_{n,m} &= F_m \Sigma_m F_m^T + F_m \Sigma_n \end{aligned} \tag{17}$$

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:

$$\begin{aligned}\hat{\Sigma}_n &= \left[\Psi_n^{-1} + \sum_{m \in C(n)} [\Sigma_{n,m}^{-1} - \Psi_n^{-1}]\right]^{-1} \\ \hat{\mu}_n &= \hat{\Sigma}_n \Big[\sum_{m \in C(n)} \Sigma_{n,m}^{-1} \mu_{n,m} \Big] \end{aligned} \tag{18}$$

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:

$$\mu'_{\text{root}} = \hat{\mu}_{\text{root}} \quad \Sigma'_{\text{root}} = \hat{\Sigma}_{\text{root}} \tag{19}$$

$$\begin{aligned}\mu'_n &= \hat{\mu}_n + J_n \left[\mu'_{\pi(n)} - \mu_{\pi(n),n}\right] \\ \Sigma'_n &= \Sigma_n + J_n \left[\Sigma'_{\pi(n)} - \Sigma_{\pi(n),n}\right] J_n^T \end{aligned} \tag{20}$$

where $J_n = \hat{\Sigma}_n F_n^T \hat{\Sigma}^{-1}_{\pi(n)}$. 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 $\mathcal{N}(\mu'_n, \Sigma'_n)$.

6. EXPERIMENTS

We demonstrate the efficacy of our model on two datasets obtained from Twitter streams. Two types of location information are attached to tweets: 1) geographical locations and 2) Twitter Places, a set of pre-defined places of interest. For geographical locations, each tweet contains a real-valued latitude and longitude vector. For Twitter Places, we convert them into real-valued latitudes and longitudes. We ignore all tweets without location information. Moreover, we remove all non-English tweets. This is achieved by a simple dictionary based method. We 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, [17, 31]). We denote this dataset as `DS1`. For this dataset, we split the tweets into **disjoint** training and test subsets such that users in the training set **do not** appear in the test set (we use 80%-20% split). 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 [17]. 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.

6.1 Hierarchies and Topics

Our model sheds some light on interesting patterns that are not easily obtained in other models: Figure 3 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).

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 2011` [31]	150.06	400
`Hong 2012` [21]	118.96	1000
`Full`	91.47	2254

Results on `DS2`	Avg. Error	Regions
`Eisenstein 2010` [17]	494	-
`Wing 2011` [30]	479	-
`Eisenstein 2011` [16]	501	-
`Hong 2012` [21]	373	100
`Full`	298	836

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. For instance, the three regions shown on the first level roughly correspond to Indonesia, the USA and the UK. This is consistent with our observation that users from these countries are active in generating geotagged Tweets. Also, by investigating the top ranked terms, we found that regional language models can capture the area dependent variations of languages, thus providing more discriminative features to location prediction. Some examples of global topics are shown in Table 1. Compared to regional language models, it is also clear that these shared topics capture higher level of interests among users across different regions[2]. Compared to similar approaches such as [21] and [16], our hierarchical structure plus global topics has more expressive power is arguably more intuitive.

6.2 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 [17] who aim to estimate a *single* location 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

[2] The general politics topic contains words like Egypt and Tunisia as these were globably popular keywords during the time the data was collected – Jan-May 2011

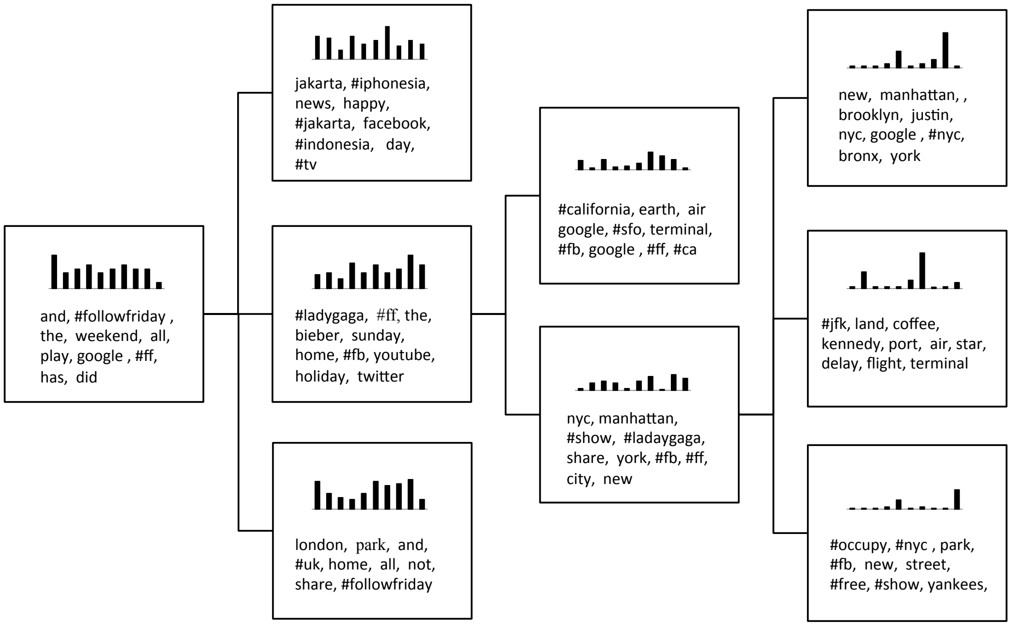

Figure 3: A small portion of the tree structure discovered from DS1.

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 2012 [21]	118.96	1000
Flat	122.43	1377
Topic	109.55	2186
Global	98.19	2034
Full	91.47	2254

Results on DS2	Avg. Error	Regions
Hong 2012 [21]	372.99	100
Flat	404.26	116
Topic	345.18	798
Global	310.35	770
Full	298.15	836

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 [13, 12] 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 denote our full model as Full.

For DS1 we compare Full with the following approaches:

Yin 2011 [31] 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 [21] 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 Figure 2.

The first observation is that Full model 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 Full, the number of regions grows naturally with the data. Also, we notice that the optimal number of regions inferred by Full is larger than its counterparts Yin 2011 and Hong 2012. We conjecture that this is because the model organizes regions in a tree-like structure and therefore more regions are needed to represent the fine scale of locations.

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

Eisenstein 2010 [17] The model is to learn a shared topic matrix 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 2011 [16] The original SAGE paper. The best reported results are used in the experiments.

Wing 2011 [30] Their method is essentially to learn regional language models per explicit regions.

Hong 2012 [21] This was the previous state of the art.

For [17, 30, 16], the authors do not report optimal regions. For [21], 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 2. 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 [21] (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 learned hierarchical structure helps the model to perform better on the prediction task.

In Section 5.3, we discussed how to sample regional language models. In Table 3 we compare the two approximation methods with directly sampling from Antoniak distributions. 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 2, we only report the results by using Minimal Paths.

6.3 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:

Flat: 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.

Topic: 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.

Global: No personal distribution over the tree structure, we assume that all tweets are generated by a fictitious user and no personal preferences are incorporated.

Full: Our full model.

The results are shown in Table 4. The first observation is that Topic, Global and Full, 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 Topic, 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 Global where no personal distributions over the tree is introduced. In summary, Full demonstrated that personalized tree structures can further boost the performance.

6.4 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 [21] on all users in the USA on DS1. The results are given in Figure 4. 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).

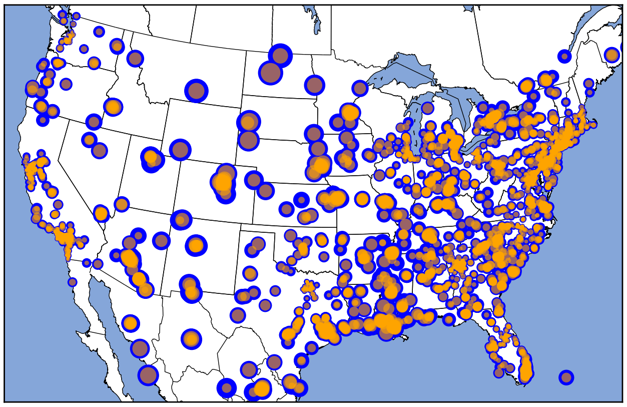

Figure 4: Error analysis for the state-of-the-art model [21] (blue) and our model (orange) on DS1.

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 [21]. This is particularly salient in the Midwest.

7. CONCLUSION

Nonparametric Bayesian models have been demonstrated as an effective tool to discover hierarchies in many applications, however, existing methods usually exhibit a global distribution over the tree structure, not allowing this distribution to vary for different users. The latter can be an impediment in applications such as behavioral targeting and online user profiling. In addition, the cascading of parameter spaces over hierarchies is usually an artifact of the stick-breaking process, not necessarily directly controlled by the design of models. This leads to inflexibility when handling multiple types of parameters which should be cascaded through the discovered tree structure. In this paper, we propose a unified framework, the nested Chinese Restaurant Franchise, to discover a *unified* hidden tree structure with unbounded width and depth while allowing users to have different distributions over this structure. Furthermore, the patterns of parameters cascading over the tree are explicitly specified. An efficient algorithm is developed to perform the posterior inference. We apply the framework to organize Twitter messages into a hierarchical structure and show that this tree structure can be used to predict locations of unlabeled messages, resulting in significant improvements to state-of-the-art approaches, as well as revealing interesting hidden patterns. For future work, we plan to exploit distributed sampling techniques and data layout as in [2, 6] in addition to hash-based sampling [5] to scale the inference algorithm to the full twitter dataset. In addition, to model temporal variation for extra location signals, we plan to extend the current work with ideas from [3, 4, 7].

Acknowledgement

We thank the anonymouse reviewers for their helpful comments and Twitter for permitting us to publish results based on dataset DS2.

8. REFERENCES

[1] R. Adams, Z. Ghahramani, and M. Jordan. Tree-structured stick breaking for hierarchical data. In *Neural Information Processing Systems*, pages 19–27, 2010.

[2] A. Ahmed, M. Aly, J. Gonzalez, S. Narayanamurthy, and A. Smola. Scalable inference in latent variable models. In *Web Science and Data Mining (WSDM)*, 2012.

[3] A. Ahmed, Q. Ho, J. Eisenstein, E. Xing, A. Smola, and C. Teo. Unified analysis of streaming news. In *Proceedings of WWW*, Hyderabad, India, 2011. IW3C2, Sheridan Printing.

[4] A. Ahmed, Q. Ho, C. H. Teo, J. Eisenstein, A. J. Smola, and E. P. Xing. Online inference for the infinite topic-cluster model: Storylines from streaming text. In *AISTATS*, 2011.

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

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

[7] A. Ahmed and E. P. Xing. Timeline: A dynamic hierarchical dirichlet process model for recovering birth/death and evolution of topics in text stream. In *UAI*, 2010.

[8] C. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Annals of Statistics*, 2:1152–1174, 1974.

[9] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The infinite hidden markov model. In *Neural Information Processing Systems*. MIT Press, 2002.

[10] D. Blackwell and J. MacQueen. Ferguson distributions via polya urn schemes. *The Annals of Statistics*, 1(2):353–355, 1973.

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

[12] Z. Cheng, J. Caverlee, K. Lee, and D. Sui. Exploring millions of footprints in location sharing services. In *International AAAI Conference on Weblogs and Social Media*, 2011.

[13] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Knowledge Discovery and Data Mining*, pages 1082–1090, New York, NY, USA, 2011. ACM.

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

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

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

[17] J. Eisenstein, B. O'Connor, N. A. Smith, and E. Xing. A latent variable model for geographic lexical variation. In *Empirical Methods in Natural Language Processing*, pages 1277–1287, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

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

[19] Q. Hao, R. Cai, C. Wang, R. Xiao, J.-M. Yang, Y. Pang, and L. Zhang. Equip tourists with knowledge mined from travelogues. In *Proceedings of WWW*, pages 401–410, New York, NY, USA, 2010. ACM.

[20] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1/2):177–196, 2001.

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

[22] L. F. James. Coag-frag duality for a class of stable poisson-kingman mixtures, 2010.

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

[24] S. Roller, M. Speriosu, S. Rallapalli, B. Wing, and J. Baldridge. Supervised text-based geolocation using language models on an adaptive grid. In *Proceedings of EMNLP*, pages 1500–1510, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[25] S. Sizov. Geofolk: latent spatial semantics in web 2.0 social media. In *Proceedings of WSDM*, pages 281–290, New York, NY, USA, 2010. ACM.

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

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

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

[29] X. Wang, A. McCallum, and X. Wei. Topical N-grams: Phrase and topic discovery, with an application to information retrieval. In *International Conference on Data Mining ICDM*, pages 697–702. IEEE Computer Society, 2007.

[30] B. Wing and J. Baldridge. Simple supervised document geolocation with geodesic grids. In *Proceedings of ACL*, pages 955–964, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

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