GMove: Group-Level Mobility Modeling Using Geo-Tagged Social Media

Chao Zhang[1], Keyang Zhang[1], Quan Yuan[1], Luming Zhang[2], Tim Hanratty[3], and Jiawei Han[1]
[1]Dept. of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA
[2]Dept. of CSIE, Hefei University of Technology, China
[2]Information Sciences Directorate, Army Research Laboratory, MD, USA
[1]{czhang82, kzhang53, qyuan, hanj}@illinois.edu [2]zglumg@gmail.com
[3]timothy.p.hanratty@mail.mil

ABSTRACT

Understanding human mobility is of great importance to various applications, such as urban planning, traffic scheduling, and location prediction. While there has been fruitful research on modeling human mobility using tracking data (*e.g.*, GPS traces), the explosively growing geo-tagged social media (GSM) brings new opportunities to this task because of its sheer size and multi-dimensional nature. Nevertheless, how to obtain quality mobility models from the highly sparse and complex GSM data remains a challenge that cannot be readily addressed by existing techniques.

We propose GMOVE, a *group-level mobility modeling* method for GSM data. Our key insight is that, the GSM data usually contains multiple user groups, where the users within the same group share significant movement regularity. Meanwhile, user grouping and mobility modeling are two intertwined tasks: (1) better user grouping offers better within-group data consistency and thus leads to more reliable mobility models; and (2) better mobility models serve as useful guidance that helps infer the group a user belongs to. GMOVE thus alternates between user grouping and mobility modeling, and generates an ensemble of Hidden Markov Models (HMMs) to characterize group-level movement regularity. Furthermore, to reduce text sparsity of GSM data, GMOVE also features a text augmenter. The augmenter computes keyword correlations by examining their spatiotemporal distributions. With such correlations as auxiliary knowledge, it performs sampling-based augmentation to alleviate text sparsity and produce high-quality HMMs.

Our extensive experiments on two real-life data sets demonstrate that GMOVE can effectively generate meaningful group-level mobility models. Moreover, with context-aware location prediction as an example application, we observe that GMOVE significantly outperforms baseline mobility models in terms of prediction accuracy.

1. INTRODUCTION

Understanding human mobility has been widely recognized as a corner-stone task for various applications, ranging from urban planning and traffic scheduling to location prediction and personalized activity recommendation. In the past few decades, the importance of this task has led to fruitful research efforts in the data mining community [6, 25, 12, 4, 24]. While most of the previous works use tracking data (*e.g.*, GPS traces) to discover human movement regularity, the recent explosive growth of *geo-tagged social media* (GSM) brings new opportunities to this task. Nowadays, as GPS-enabled mobile devices and location-sharing apps are penetrating into our daily life, every single day is witnessing millions of geo-tagged tweets, Facebook checkins, Instagram posts, *etc.* Compared with conventional tracking data, GSM possesses two unique advantages for mobility modeling: (1) First, in addition to spatial and temporal information, each GSM record also includes text. Hence, a user's GSM history reveals not only how she moves from one location to another, but also what activities she does at different locations. (2) Second, the GSM data has a much larger size and coverage. Due to privacy concerns, gathering tracking data typically requires a huge amount of time and money to engage sufficient volunteers. In contrast, the GSM data naturally covers hundreds of millions of users and has a much larger volume.

Considering its multi-dimensional nature and sheer size, it is clear that GSM serves as an unprecedentedly valuable source for human mobility modeling. Hence, we study the problem of using large-scale GSM data to unveil human movement regularity. Specifically, we answer the following two questions:

- *What are the intrinsic states underlying people's movements?* To name a few, a state could be working at office in the morning, exercising at gym at noon, or having dinner with family at night. We want each state to provide a unified 3W view regarding a user's activity: (1) where is the user; (2) what is the user doing; and (3) when does this activity happen.

- *How do people move sequentially between those latent states?* For example, where do people working in Manhattan usually go to relax after work? and what are the popular sightseeing routes for a one-day trip in Paris? We aim to summarize people's transitions between the latent states in a concise and interpretable way.

Unveiling human mobility using GSM data can greatly benefit various applications. Consider traffic jam prevention as an example. While existing techniques [6, 24, 23] can detect sequential patterns to reflect the populace flow between geographical regions, the GSM data allows us understand populace flow beyond that. Suppose severe traffic jams occur frequently in a region R. With the mobility model learnt from massive GSM data, we can understand what are people's typical activities in R, which regions do people come from, and why do people take those trips. Such understandings can guide the government to better diagnose the root causes of traffic jams and take prevention actions accordingly. Another

example application is context-aware location prediction. Previous studies [14, 13, 4] typically use the spatiotemporal regularity of human mobility to predict the next location a person may visit. By leveraging GSM data, we can incorporate activity-level transitions to infer what activities a user will engage subsequently. As such, the mobility models learnt from GSM can potentially improve context-aware location prediction remarkably.

Despite its practical importance, the task of modeling human mobility with GSM data is nontrivial due to several challenges: (1) *Integrating diverse types of data.* The GSM data involves three different data types: location, time, and text. Considering the totally different representations of those data types and the complicated correlations among them, how to effectively integrate them for mobility modeling is difficult. (2) *Constructing reliable models from sparse and complex data.* Unlike intentionally collected tracking data, the GSM data is low-sampling in nature, because a user is unlikely to report her activity at every visited location. Such data sparsity makes it unrealistic to train a mobility model for each individual. On the other hand, one may propose to aggregate the data of all users to train one single model, but the obtained model could then suffer from severe data inconsistency because different users can have totally different moving behaviors. (3) *Extracting interpretable semantics from short GSM messages.* The GSM messages are usually short. For example, any geo-tagged tweets contain no more than 140 characters, and most geo-tagged Instagram photos are associated with quite short text messages. It is nontrivial to extract reliable knowledge from short GSM messages and build high-quality mobility models.

Contributions. We propose GMOVE, an effective method that models human mobility from massive GSM data. GMOVE relies on the Hidden Markov Model (HMM) to learn the multi-view latent states and people's transitions between them. To obtain quality HMMs from the highly sparse and complex GSM data, we propose a novel idea of *group-level* mobility modeling. The key is to group the users that share similar moving behaviors, *e.g.*, the students studying at the same university. By aggregating the movements of like-behaved users, GMOVE can largely alleviate data sparsity without compromising the within-group data consistency.

To achieve effective user grouping and mobility modeling, we find that these two sub-tasks can mutually enhance each other: (1) better user grouping offers better within-group data consistency, which helps produce more reliable mobility models; and (2) better mobility models can serve as useful knowledge, which helps infer the group a user belongs to. Based on this observation, we develop an iterative framework, in which we alternate between user grouping and group-level mobility modeling. We theoretically and empirically show that, such a process leads to better user grouping and mobility models after each iteration, and finally generates an ensemble of high-quality HMMs.

Another important component of GMOVE is a text augmenter, which leverages keyword spatiotemporal correlations to reduce text sparsity. While each raw GSM message is short and noisy, we find that using millions of GSM records, we are able to extract the intrinsic correlations between keywords by examining their spatiotemporal distributions. Consider two users who are having dinner at the same Italian restaurant and posting tweets with two different keywords: "pasta" and "pizza". Although these two keywords do not co-occur in the same tweet, they are spatially and temporally close. We hence quantify the spatiotemporal correlations between keywords, and use such correlations as auxiliary knowledge to augment raw GSM messages. We show that such augmentation can largely reduce text sparsity and lead to highly interpretable states.

To summarize, we make the following contributions:

1. We formulate the problem of mobility modeling using massive geo-tagged social media (Section 2). To the best of our knowledge, this is the first study that attempts to use the multi-dimensional (spatial, temporal, textual) information in GSM data to unveil human movement regularity.

2. We propose a novel group-level mobility modeling framework. Built upon an ensemble of HMMs (Section 4), it enables user grouping and mobility modeling to mutually enhance each other. As such, it effectively alleviates data sparsity and inconsistency to generate reliable mobility models.

3. We design a strategy that leverages the spatiotemporal distributions to quantify keyword correlations as auxiliary knowledge (Section 3). By augmenting raw GSM messages with such knowledge, our method largely reduces text sparsity to produce interpretable latent states.

4. We perform extensive experiments using two real data sets (Section 5). The results show that our method can produce high-quality group-level mobility models. Meanwhile, with context-aware location prediction as an example application, we observe that our method achieves much better prediction accuracy compared with baseline mobility models.

2. PRELIMINARIES

In this section, we formulate the problem of mobility modeling using geo-tagged social media, and explore several characteristics of this problem to motivate the design of GMOVE.

2.1 Problem Description

A GSM record x is defined as a tuple $\langle u_x, t_x, l_x, e_x \rangle$ where: (1) u_x is the user id; (2) t_x is the creating timestamp (in second); (3) l_x is a two-dimensional vector representing the user's location when x is created; and (4) e_x, which is a bag of keywords from a vocabulary V, denotes the text message of x.

Let us consider a set of users $U = \{u_1, u_2, \ldots, u_M\}$. For a user $u \in U$, her GSM history is a sequence of chronologically ordered GSM records $S_u = x_1 x_2 \ldots x_n$. To understand u's mobility, one may propose to use the entire sequence S_u. Nevertheless, the sequence S_u is low-sampling in nature, and the temporal gaps between some pairs of consecutive records can be very large, *e.g.*, several days. To generate reliable mobility models, we only use the dense parts in S_u, which we define as *movements*.

DEFINITION 1 (MOVEMENT). *Given a GSM sequence $S = x_1 x_2 \ldots x_n$ and a time gap threshold $\Delta t > 0$, a subsequence $S' = x_i x_{i+1} \ldots x_{i+k}$ is a length-k movement of S if S' satisfies: (1) $\forall 1 < j \leq k$, $t_{x_j} - t_{x_{j-1}} \leq \Delta t$; and (2) there are no longer subsequences in S that contains S' and meanwhile also satisfies condition (1).*

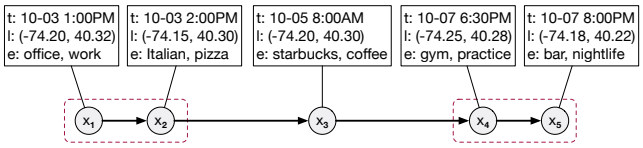


Figure 1: An illustration of *movement* ($\Delta t = 3$ hours).

EXAMPLE 1. *In Figure 1, given a user's GSM history $S = x_1 x_2 x_3 x_4 x_5$ and $\Delta t = 2$ hours, there are two movements: $x_1 x_2$ and $x_4 x_5$. These movements correspond to the user's two moving behaviors: (1) having dinner at an Italian restaurant after work; and (2) going to bar after taking exercises.*

A movement essentially leverages the time constraint to extract reliable moving behaviors in the GSM history. With the extracted movements, our general goal is to use them to understand the mobility of the users in U. Nevertheless, the number of movements of one single user is usually too limited to train a mobility model, while aggregating the movements of all the users in U could suffer from severe data inconsistency and result in an ambiguous model.

To address the dilemma, we propose to study *group-level user mobility*. Our key observation is that, aggregating the movements of like-behaved users can largely alleviate data sparsity and inconsistency, and thus unveil the movement regularity for that group. Consider a group of students studying at the same university, they may share similar hotspots like classrooms, gyms, and restaurants. Many students also have similar moving behaviors, *e.g.*, moving from the same residence hall to classrooms, and having lunch together after classes. As another example, a group of tourists in New York may all visit locations like the JFK Airport, the Metropolitan Museum, the 5th Avenue, *etc.*. While the data of one individual is limited, the collective movement data from a group of like-behaved people can be informative about their latent hotspots and the movement regularity. It is important to note that, one user can belong to several different groups, *e.g.*, a student may also go sightseeing in the city on weekends. Therefore, the users should be grouped softly instead of rigidly.

Based on the above observation, we formulate the problem of *group-level mobility modeling* as follows. Given the set U of users and their movements extracted from historical GSM records, the task of group-level mobility modeling consists of two sub-tasks: (1) ***user grouping***: softly group the users in U such that the members in the same group have similar moving behaviors; and (2) ***mobility modeling***: for each user group, discover the latent states that reflect the users' activities from a multidimensional (where-what-when) view, and find out how the users move between those latent states.

2.2 Overview of GMOVE

GMOVE is built upon the Hidden Markov Model (HMM), an effective statistical model for sequential data. Given an observed sequence $S = x_1 x_2 \dots x_n$, HMM defines K latent states $Z = \{1, 2, \dots, K\}$ that are not observable. Each record $x_i (1 \leq i \leq n)$ corresponds to a latent state $z_i \in Z$, which has a probabilistic distribution that governs the generation of x_i. In addition, the latent state sequence $z_1 z_2 \dots z_n$ follows the Markov process, namely, the state z_i only depends on the previous state z_{i-1}.

As discussed above, there exists a dilemma when applying HMM to understand the mobility of the users in U: training an HMM for each user is unrealistic due to data sparsity, while training one single HMM for all the users leads to an ambiguous model due to data inconsistency. Therefore, GMOVE performs group-level mobility modeling by coupling the subtasks of user grouping and mobility modeling. It aims at dividing the users in U into like-behaved user groups and training an HMM for each group. By aggregating the movement data within the same group, we can alleviate data sparsity without compromising data consistency.

Now the question is, how can we group the users such that the users within the same group have similar moving behaviors? Our idea is that the sub-tasks of *user grouping* and *mobility modeling* can mutually enhance each other: (1) better user grouping leads to more consistent movement data within each group, which can improve the quality of the HMM; and (2) high-quality HMMs more accurately describe the true latent states and the transitions between them, which helps infer which group a user should belong to. Hence, GMOVE employs an iterative framework that alternates between user grouping and HMM training. Later we will see, as the iteration continues, the user grouping gets more and more accurate, and the mobility models get more and more reliable, and such a process will finally converge to a stable solution.

Another important component of GMOVE is a text augmenter. When training an HMM for each group, we model each latent state to generate multidimensional (spatial, temporal, and textual) observations of the user's activity. While the spatial and temporal observations are clean, the major challenge in obtaining high-quality latent states is that the text messages are mostly short and noisy. To address this problem, we observe that keywords' spatiotemporal distributions can reveal their intrinsic correlations. The text augmenter thus quantifies the spatiotemporal correlations between keywords to obtain auxiliray knowledge. With such knowledge, it performs a weighted sampling process to augment raw text messages. Such augmentation can largely overcome text sparsity and suppress noise to help produce high-quality HMMs.

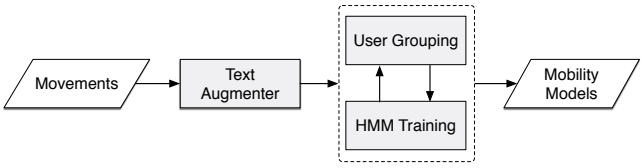


Figure 2: The framework of GMOVE.

Figure 2 summarizes the framework of GMOVE. Given the input movements, GMOVE consists of two major modules to produce group-level mobility models. The first module is the text augmenter, which examines the spatiotemporal correlations between keywords to augment raw messages. With the augmented movements, the second module obtains an ensemble of HMMs by alternating between user grouping and HMM training. In the following, we elaborate these two module in Section 3 and 4, respectively.

3. TEXT AUGMENTATION

In this section, we describe GMOVE's text augmenter, which first computes keyword correlations and then performs sampling-based text augmentation.

Keyword correlation computation. Our computation of keyword correlation relies on spatiotemporal discretization. Let us consider a spatiotemporal space D that consists of three dimensions: (1) the longitude dimension x; (2) the latitude dimension y; and (3) the time dimension t. We partition D into $N_x \times N_y \times N_t$ equal-size grids, where N_x, N_y and N_t are pre-specified integers that control discretization granularity. Based on such discretization, we define the concepts of *grid density* and *signature* for each keyword.

DEFINITION 2 (GRID DENSITY). *Given a keyword w, the density of w in grid $\langle n_x, n_y, n_t \rangle$ $(1 \leq n_x \leq N_x, 1 \leq n_y \leq N_y, 1 \leq n_t \leq N_t)$ is defined as w's frequency in that grid, namely*

$$d_w(n_x, n_y, n_t) = \frac{c_w(n_x, n_y, n_t)}{\sum_{n_x, n_y, n_t} c_w(n_x, n_y, n_t)},$$

where $c_w(n_x, n_y, n_t)$ is the number of GSM records that contain w and meanwhile fall in grid $\langle n_x, n_y, n_t \rangle$.

DEFINITION 3 (SIGNATURE). *Given a keyword w, its signature s_w is a $N_x N_y N_t$-dimensional vector, where $d_w(n_x, n_y, n_t)$ is the value for the $((n_t - 1)N_x N_y + (n_y - 1)N_x + n_x)$-th dimension.*

The signature of a keyword encodes how frequently that keyword appears in different spatiotemporal grids. Intuitively, if two keywords are semantically correlated (*e.g.*, "pasta" and "pizza"), they are more likely to frequently co-occur in the same grid and

thus have similar signatures. Below, we measure the spatiotemporal correlation between two keywords by simply computing the similarity of their signatures.

DEFINITION 4 (KEYWORD CORRELATION). *Given two keywords w_i and w_j, let s_{w_i} and s_{w_j} be their signatures. The spatiotemporal correlation between w_i and w_j, denoted as $corr(w_i, w_j)$, is the cosine distance between s_{w_i} and s_{w_j}.*

Sampling-based augmentation. Definition 4 quantifies keyword correlations based on their spatiotemporal distributions. With Definition 4, we further select out a set of vicinity keywords from the vocabulary for each keyword.

DEFINITION 5 (VICINITY). *For any keyword $w \in V$, its vicinity $\mathcal{N}_w$ is $\mathcal{N}_w = \{v | v \in V \wedge corr(w, v) \geq \delta\}$ where $\delta \geq 0$ is a pre-specified correlation threshold.*

The vicinity concept is important as it identifies other keywords that are highly correlated to the target keyword and suppresses noisy keywords. Based on Definition 5, we are now ready to describe the text augmentation process. Given an input GSM record x, our goal is to augment the original text message e_x by incorporating many other keywords that are semantically relevant to e_x. As shown in Algorithm 1, the augmentation simply performs sampling by using keyword correlation as auxiliary knowledge. Specifically, we first sample a keyword w from the original message based on normalized TF-IDF weights, and then sample a relevant keyword v from w's vicinity. The probability of a keyword v being sampled is proportional to its spatiotemporal correlation with w, namely $corr(w, v)$. We repeat the sampling process until the augmented message reaches a pre-specified length L.

Algorithm 1: Text augmentation.

Input: A GSM record x, the target length L.
Output: The augmented text message of x.

$A_x \leftarrow$ The original text message e_x;
while $len(A_x) < L$ **do**
  Sample a word $w \in e_x$ with probability $\frac{TF-IDF(w)}{\sum_{v \in e_x} TF-IDF(v)}$;
  Sample a word $v \in \mathcal{N}_w$ with probability $\frac{corr(w,v)}{\sum_{u \in \mathcal{N}_w} corr(w,u)}$;
  Add v into A_x;
return A_x;

4. HMM ENSEMBLE LEARNING

In this section, we describe GMOVE's second module that learns an ensemble of HMMs. Below, we first present an iterative refinement framework in Section 4.1, Then we introduce the procedures for the HMM training and user grouping in Section 4.2 and 4.3, respectively. Finally, we prove the convergence of GMOVE and analyze its time complexity in Section 4.4.

4.1 The Iterative Refinement Framework

To generate high-quality group-level HMMs, GMOVE employs an iterative refinement framework that performs the following steps:

1. **Initialization:** Let $U = \{u_1, u_2, \ldots, u_M\}$ be the user set, and $\mathcal{G} = \{1, 2, 3, \ldots, G\}$ be the G underlying user groups.
   (a) $\forall u \in U$, randomly generate an initial membership vector M_u *s.t.* $\sum_{g=1}^{G} M_u(g) = 1$, where $M_u(g)$ denotes the probability that user u belongs to group g.
   (b) $\forall g \in \mathcal{G}$, randomly initialize an HMM H_g. The ensemble of the HMMs are denoted as $\mathcal{H} = \{H_g | g \in \mathcal{G}\}$.
2. **HMM Training:** $\forall g \in \mathcal{G}$, use the membership vectors to reweigh all input movements, such that the weights of user u's movements are set to $M_u(g)$. Then refine every H_g to generate a new HMM ensemble, $\mathcal{H}^{\text{new}} = \{H_g^{\text{new}} | g \in \mathcal{G}\}$.
3. **User Grouping:** $\forall u \in U$, use $\mathcal{H}^{\text{new}}$ to update u's membership vector such that the g-th dimension is the posterior probability that user u belongs to group g, namely $M_u^{\text{new}}(g) = p(g|u; \mathcal{H}^{\text{new}})$.
4. **Iterating:** Check for convergence using the log-likelihood of the input movements. If the convergence criterion is not met, then let
$$\forall g, H_g \leftarrow H_g^{\text{new}}; \ \forall u, M_u \leftarrow M_u^{\text{new}};$$
and return to Step 2.

4.2 HMM Training

In the HMM training step, the task is to use weighted movements to train a new HMM for each group. Below, we first define the HMM for generating the movements, and then discuss how to infer model parameters.

4.2.1 Group-Level HMM

Let us consider a movement $S = x_1 x_2 \ldots x_N$. Each observation $x_n (1 \leq n \leq N)$ is multidimensional: (1) l_n is a two-dimensional vector denoting latitude and longitude; (2) t_n is a timestamp; and (3) e_n is a bag of keywords denoting the augmented message.

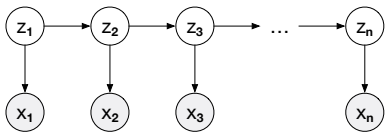


Figure 3: An illustration of the HMM.

To generate the sequence $x_1 x_2 \ldots x_N$, we assume there are K latent states $Z = \{1, 2, \ldots, K\}$. As shown in Figure 3, each observation x_n corresponds to a latent state $z_n \in Z$, and the sequence of the latent states $z_1 z_n \ldots z_N$ follows a Markov process parameterized by: (1) a K-dimensional vector π that defines the initial distribution over the K latent states, *i.e.*, $\pi_k = p(z_1 = k)$ $(1 \leq k \leq K)$; and (2) a $K \times K$ matrix $\mathbf{A}$ that defines the transition probabilities among the K latent states. Suppose the $(n-1)$-th $(n > 1)$ latent sate is $z_{n-1} = j$, then the n-th state z_n depends only on z_{n-1}. More formally, if the state z_{n-1} is j, the probability for the state z_n to be k is given by $\mathbf{A}_{jk}$, *i.e.*, $p(z_n = k | z_{n-1} = j) = \mathbf{A}_{jk}$.

Meanwhile, a state z_n generates x_n by generating the location l_n, the timestamp t_n, and the keywords e_n independently, *i.e.*,

$$p(x_n | z_n = k) = p(l_n | z_n = k) \cdot p(t_n | z_n = k) \cdot p(e_n | z_n = k).$$

For each observation x_n, we assume the following spatial, temporal, and textual distributions: (1) The location l_n is generated from a bi-variate Gaussian, *i.e.*, $p(l_n | z_n = k) = N(l_n | \mu_{lk}, \Sigma_{lk})$ where μ_{lk} and Σ_{lk} are the mean and variance matrix for state k. (2) The timestamp t_n is generated from a uni-variate Gaussian, *i.e.*, $p(t_n | z_n = k) = N(t_n \mod 86400 | \mu_{tk}, \sigma_{tk})$, where μ_{tk} and σ_{tk} are the mean and variance for state k. Note that we convert the raw absolute timestamp t_n (in second) into the relative timestamp in one day. (3) The keywords e_n are generated from a multinomial distribution, *i.e.*, $p(e_n | z_n = k) \propto \prod_{v=1}^{V} \theta_{kv}^{e_n^v}$, where θ_{kv} is the probability of choosing word v for state k, and e_n^v is the number of v's occurrences in e_n.

4.2.2 Parameter Inference

When training the HMM for group g, let us use $\{S_1, S_2, \ldots, S_R\}$ to denote the input movements, and each movement $S_r (1 \leq r \leq R)$ is associated with a weight w_r, denoting the probability that the user of S_r belongs to group g. The group-level HMM is parameterized by $\Phi = \{\pi, \mathbf{A}, \mu_l, \Sigma_l, \mu_t, \sigma_t, \theta\}$. We use the Expectation-Maximization (EM) algorithm to estimate these parameters. It is important to note that, when applying EM to train the HMM for group g, we use the previous HMM H_g to initialize model parameters. Later we will see, such an initialization ensures the iterative framework of GMOVE obtains better HMMs after each iteration and finally converge.

Starting with the initial parameters, the EM algorithm alternates between the E-step and the M-step round by round[1] to generate a new HMM. In the $(t+1)$-th round E-step, it utilizes the estimated parameters at the t-th round $\Phi^{(t)}$, and computes the expectation of the complete likelihood $Q(\Phi)$. In the M-step, it finds a new estimation $\Phi^{(t+1)}$ that maximizes the Q-function. Below, we present the details of the E-step and the M-step.

E-Step. In the E-step, the key is to compute the distribution of the latent states based on the old parameter set $\Phi^{(t)}$. Given a movement $S_r = x_{r,1}x_{r,2}\ldots x_{r,N}$, $\forall 1 \leq n \leq N$, we first use the Baum-Welch algorithm to compute two distributions:

$$\alpha(z_{r,n}) = p(x_{r,1}, x_{r,2}, \ldots, x_{r,n}, z_{r,n}; \Phi^{(t)}),$$
$$\beta(z_{r,n}) = p(x_{r,n+1}, \ldots, x_{r,N} | z_{r,n}; \Phi^{(t)}).$$

Here, $\alpha(z_{r,n})$ can be computed in a forward fashion, and $\beta(z_{r,n})$ can be computed in a backward fashion:

$$\alpha(z_{r,n}) = p(x_{r,n}|z_{r,n}) \sum_{z_{r,n-1}} \alpha(z_{r,n-1}) p(z_{r,n}|z_{r,n-1}),$$
$$\beta(z_{r,n}) = \sum_{z_{r,n+1}} \beta(z_{r,n+1}) p(x_{r,n+1}|z_{r,n+1}) p(z_{r,n+1}|z_{r,n}).$$

where the initial distributions $\alpha(z_{r,1})$ and $\beta(z_{r,N})$ are:

$$\alpha(z_{r,1} = k) = \pi_k p(x_{r,1}|z_{r,1} = k), \;\; \beta(z_{r,N} = k) = 1.$$

Based on $\alpha(z_{r,n})$ and $\beta(z_{r,n})$, we are now able to compute the following distributions for the latent states: (1) $\gamma(z_{r,n})$: the distribution of the n-th latent state for the r-th movement; and (2) $\xi(z_{r,n-1}, z_{r,n})$: the joint distribution of two consecutive latent states in the r-th movement. These two distributions are given by

$$\gamma(z_{r,n}) = p(z_{r,n}|S_r) = \alpha(z_{r,n})\beta(z_{r,n})/p(S_r),$$
$$\xi(z_{r,n-1}, z_{r,n}) = p(z_{r,n-1}, z_{r,n}|S_r)$$
$$= \alpha(z_{r,n-1}) p(x_{r,n}|z_{r,n}) p(z_{r,n}|z_{r,n-1}) \beta(z_{r,n})/p(S_r),$$

where $p(S_r) = \sum_{z_{r,N}} \alpha(z_{r,N})$.

M-Step. In the M-step, we derive the best estimation for the parameter set $\Phi^{(t+1)} = \{\pi, \mathbf{A}, \mu_g, \Sigma_g, \mu_t, \sigma_t, \theta\}$. Based on $\gamma(z_{r,n})$ and $\xi(z_{r,n-1}, z_{r,n})$, let us define

$$\Gamma_k = \sum_{r=1}^{R}\sum_{n=1}^{N} w_r \gamma(z_{r,n}^k); \; \Xi_j = \sum_{r=1}^{R}\sum_{n=2}^{N}\sum_{i=1}^{K} w_r \xi(z_{r,n-1}^j, z_{r,n}^i).$$

We update the parameters in $\Phi^{(t+1)}$ as follows (please see Appendix for the detailed derivation):

$$\pi_k^{(t+1)} = \sum_{r=1}^{R} w_r \gamma(z_{r,1}^k); \tag{1}$$

$$A_{jk}^{(t+1)} = \frac{1}{\Xi_j} \sum_{r=1}^{R}\sum_{n=2}^{N} w_r \xi(z_{r,n-1}^j, z_{r,n}^k) \tag{2}$$

$$\mu_{lk}^{(t+1)} = \frac{1}{\Gamma_k} \sum_{r=1}^{R}\sum_{n=1}^{N} w_r \gamma(z_{r,n}^k) l_{r,n} \tag{3}$$

$$\Sigma_{lk}^{(t+1)} = \frac{1}{\Gamma_k} \sum_{r=1}^{R}\sum_{n=1}^{N} w_r \gamma(z_{r,n}^k)(l_{r,n} - \mu_{lk}^{(t+1)})(l_{r,n} - \mu_{lk}^{(t+1)})^T; \tag{4}$$

$$\mu_{tk}^{(t+1)} = \frac{1}{\Gamma_k} \sum_{r=1}^{R}\sum_{n=1}^{N} w_r \gamma(z_{r,n}^k) t_{r,n}; \tag{5}$$

$$\sigma_{tk}^{(t+1)} = \frac{1}{\Gamma_k} \sum_{r=1}^{R}\sum_{n=1}^{N} w_r \gamma(z_{r,n}^k)(t_{r,n} - \mu_{tk}^{(t+1)})^2; \tag{6}$$

$$\theta_{kv}^{(t+1)} = \frac{1}{\Gamma_k} \sum_{r=1}^{R}\sum_{n=1}^{N} w_r \gamma(z_{r,n}^k) e_{r,n}^v. \tag{7}$$

4.3 User Grouping

Once the new ensemble of HMMs, $\mathcal{H}^{\text{new}}$, are obtained, we use them to softly assign each user into the G groups. For each user u, we update the membership vector M_u such that the g-th dimension is the posterior probability that u belongs to group g. Let us denote the set of u's movements as J_u, in which the j-th ($1 \leq j \leq |J_u|$) movement is S_u^j. Based on the new HMM ensemble $\mathcal{H}^{\text{new}}$, the probability of observing S_u^j given group g, $p(S_u^j|g; \mathcal{H}^{\text{new}})$, can be computed using the forward scoring algorithm of HMM. Accordingly, the probability of observing J_u is given by:

$$p(u|g; \mathcal{H}^{\text{new}}) = \prod_{j=1}^{|J_u|} p(S_u^j|g; \mathcal{H}^{\text{new}}). \tag{8}$$

Using the Bayes' theorem, we further derive the posterior probability that user u belongs to group g as follows:

$$p(g|u; \mathcal{H}^{\text{new}}) \propto p(g) p(u|g; \mathcal{H}^{\text{new}}),$$

where $p(u|g; \mathcal{H}^{\text{new}})$ is given by Equation 8, and $p(g)$ is estimated from the membership vectors: $p(g) = \sum_{u \in U} M_u(g)/|U|$. Finally, we obtain the new membership vector M_u^{new} as $M_u^{\text{new}}(g) = p(g|u; \mathcal{H}^{\text{new}})$.

4.4 Discussions

Now we proceed to prove the convergence of GMOVE.

THEOREM 1. *The iterative refinement framework of* GMOVE *is guaranteed to converge.*

PROOF. With Jensen's inequality, the log-likelihood of the input movement data satisfies:

$$l(\mathcal{H}) = \sum_{u \in U} \log \sum_{g=1}^{G} p(u, g; \mathcal{H}) \tag{9}$$

$$= \sum_{u \in U} \log \sum_{g=1}^{G} M_u(g) \frac{p(u, g; \mathcal{H})}{M_u(g)} \tag{10}$$

$$\geq \sum_{u \in U} \sum_{g=1}^{G} M_u(g) \log \frac{p(u, g; \mathcal{H})}{M_u(g)} \tag{11}$$

[1] We intentionally use two different terms, namely *iteration* and *round*, to distinguish the two different iterative processes: (1) GMOVE's iterative refinement framework; and (2) the EM algorithm for HMM training.

Recall that, after each iteration, GMOVE sets $M_u(g)$ to the posterior probability $p(g|u;\mathcal{H})$. The quantity $p(u,g;\mathcal{H})/M_u(g)$ is thus a constant, and hence the equality in Equation 11 is guaranteed to hold. Such a property allows us to prove the log-likelihood is non-decreasing after each iteration. More formally, we have

$$l(\mathcal{H}) = \sum_{u\in U}\sum_{g=1}^{G} M_u^t(g)\log\frac{p(u,g;H_g)}{M_u^t(g)} \quad (12)$$

$$\leq \sum_{u\in U}\sum_{g=1}^{G} M_u^t(g)\log\frac{p(u,g;H_g^{\text{new}})}{M_u^t(g)} \leq l(\mathcal{H}^{\text{new}}) \quad (13)$$

In the above, the key step is Equation 13. It holds because in its iterative refinement framework, GMOVE uses the parameters of the previous HMM ensemble to initialize the current HMM ensemble, which guarantees $p(u,g;\mathcal{H})$ to be non-decreasing after the HMM refinement. As the total likelihood is non-decreasing after every iteration, we have proved the convergence of our algorithm. □

There is a tight connection between our iterative refinement framework and the traditional EM algorithm. The only difference between the two is that, in the EM algorithm, the constructed lower bound is typically convex and thus can be easily optimized to global optimum; while in our case, the constructed lower bound is still non-convex and we need to use yet another EM algorithm (HMM refinement) to optimize this lower bound. As the lower bound is non-convex, we have to make sure the HMM training of every iteration initialize with the parameters learnt from the previous iteration, otherwise we may risk running into different local optimums and thus break the convergence guarantee of our algorithm.

Time complexity. As GMOVE consists of two modules: text augmentation and HMM ensemble learning, we analyze their time cost separately. For text augmentation, we first need to compute the correlations between all pairs of keywords based on their spatiotemporal signatures, which results in a time complexity of $O(|V|^2 N_x N_y N_t)$. After computing keyword correlations, we perform a sampling process to augment each GSM record. Assume the total number of GSM records in the movement data is E, then the cost of the sampling process is simply $O(EL)$. Hence, the overall time complexity of text augmentation is $O(|V|^2 N_x N_y N_t + EL)$. For the HMM ensemble learning module, let N_1 be the maximum number of iterations in GMOVE's refinement framework, and N_2 be the maximum number of rounds for training one HMM. Then the time complexity of HMM ensemble learning is $O(N_1 N_2 GE(K^2 + KL))$.

5. EXPERIMENTS

In this section, we evaluate the empirical performance of GMOVE. All the algorithms were implemented in JAVA and the experiments were conducted on a computer with Intel Core i7 2.4Ghz CPU and 8GB memory.

5.1 Data Sets

Our experiments are based on two real-life data sets, both of which are crawled using Twitter Streaming API[2] during 2014.10.10 — 2014.10.30. The first data set, referred to as LA, consists of 0.6 million geo-tagged tweets in Los Angeles. After grouping the tweets by user id and extracting movements with a time constraint of three hours (Definition 1), we obtain around 30 thousand movements. The second data set, referred to as NY, consists of 0.7 million geo-tagged tweets in New York, and there are 42 thousand movements after extracting with the three-hour time constraint. The reason we choose these two cities is because we want to verify the performance of GMOVE on the data sets that have quite different population distributions — the populace of Los Angeles is spread out in many different districts, while the populace of New York is relatively concentrated in Manhattan and Brooklyn.

[2]https://dev.twitter.com/streaming/overview

5.2 Illustrative Cases

In this subsection, we run GMOVE on LA and NY and use several illustrative cases to verify its effectiveness.

5.2.1 Text Augmentation Results

In the first set of experiments, we demonstrate the effectiveness of the text augmentation module of GMOVE. To this end, we randomly choose several tweets in both LA and NY, and use GMOVE to augment the raw tweet messages. Table 1 shows the augmentation results for four example tweets (two from LA and two from NY). As a preprocessing step, we performed stemming and stopwords removal for the raw tweets, and also discretize the spatiotemporal space by partitioning each dimension into ten bins. When augmenting each tweet, the correlation threshold is set to $\delta = 0.2$ and the augmentation size is set to $L = 100$.

Table 1: Text augmentation examples on LA and NY. The number beside each word in the augmented message denotes that word's number of occurrences in the augmented message.

Data	Raw tweet message	Augmented message
LA	Y'all just kobe fans not lakers. Let's go lakers!!	fans(11), game(7), kobe(19), jeremy(6), lakers(26), injury(8), staples(8), center(4), nba(9), bryant(12)
	Fun night! @ Universal Studios Hollywood http://t.co/wMibfyleTW	fun(4), universal(20), studio(16), hollywood(18), night(5), party(7), fame(6), people(13), play(11)
NY	Nothing better...fresh off the oven! #Italian #bakery #pizza	fresh(7), oven(21), italian(19), bakery(12), pizza(14), bread(6), cook(5), food(12), kitchen(4)
	My trip starts now! @ JFK Airport	jfk(24), international(5), trip(9), travel(6), john(13), kennedy(14), terminal(8), start(6), now(3), airport(12)

Examining Table 1, we find that GMOVE's text augmenter is quite effective: given the short raw tweets, GMOVE generates semantically comprehensive messages by including relevant keywords that are not mentioned in the raw tweets. Consider the first tweet as an example. A fan of the Lakers (a basketball team) was posting to support his team. The original tweet message contains only three meaningful keywords: "kobe", "lakers", and "fan". By computing the spatiotemporal correlations between different keywords, GMOVE successfully identifies several other keywords (*e.g.*, "game", "staples", "center") that are relevant to the three seed keywords. The augmented tweet becomes much more semantically comprehensive, making it possible to connect with other Lakers-related tweets to form high-quality latent states. Similar phenomena are also observed on the other three tweets.

5.2.2 Group-Level Mobility Models

In this set of experiments, we exemplify the group-level mobility models generated by GMOVE on LA and NY. We are interested in the following questions: (1) can GMOVE indeed discover high-quality latent states and meaningful transitions among them? and (2) do different group-level mobility models reflect different mobility patterns? To answer the above questions, we ran GMOVE on LA and NY with the following parameter settings: $\delta = 0.2$, $L = 100$,

$G = 80$, and $K = 10$. While GMOVE discovers 80 group-level mobility models on both LA and NY, we choose two representative mobility models learnt on LA due to space limit. In Figure 4(a) and 4(b), we visualize those two example mobility models.

Based on the keywords and locations of the latent states, we can say the mobility model in Figure 4(a) likely corresponds to the movements of a group of UCLA students. This is because most latent states are centered around the UCLA campus, and meanwhile carry college-life semantics (*e.g.*, home, school, gym, friends). We have the following interesting findings from the mobility model: (1) *First, the latent states are highly interpretable.* For instance, the top keywords of state 1 are "school", "ucla", "course", *etc.*. It clearly reflects the students' on-campus activities, such as attending classes, taking exams, and staying with friends. (2) *Second, the transitions between latent states are practically meaningful.* Still consider state 1 as an example. As shown by the transition matrix, after taking classes at campus, the top subsequent activities are going back home (state 0 & 8), having dinner (state 2 & 6), and doing sports (state 7). Those transitions are all highly consistent with a student's lifestyle in real life, and thus reflects the high quality of the underlying group-level HMM.

The mobility model shown in Figure 4(b) probably corresponds to the activities of tourists in LA. Again, we find that the latent states (*e.g.*, airport, hotel, Hollywood, beach) are highly interpretable and the transitions (*e.g.*, going to hotel from airport, having food after sightseeing) are intuitive. More importantly, when comparing the mobility models in Figure 4(a) and 4(b), one can clearly observe that these two models reflect totally different hotspots and moving behaviors. This phenomenon suggests that GMOVE can indeed effectively partition the users based on their behaviors, and generate highly descriptive HMMs for different groups.

5.3 Quantitative Evaluation

While GMOVE can be used for various real-life applications, we choose *context-aware location prediction* as an example to quantitatively evaluate the performance of GMOVE.

5.3.1 Experimental Setup

Task description. Context-aware location prediction aims at predicting the next location of a user, given the current observation of the user's movement. Formally, given a ground-truth movement $x_1 x_2 \dots x_N$, we assume the sequence $x_1 x_2 \dots x_{N-1}$ is already known, and try to recover x_N from a pool of candidate locations. The candidate pool, denoted as $\mathcal{C}_x$ is formed as follows: from all the geo-tagged tweets, we select the ones whose geographical distance to x_N is no larger than three kilometers and the temporal difference to x_N is no larger than five hours. Once the candidate pool is obtained, we use the given mobility model to rank all the candidates in the descending order of visiting probability, and check whether the grouth-truth record x_N appears in the top-k results (k is an integer that controls the result list size).

Intuitively, the better a mobility model is, the more likely it can recover the ground-truth record x_N (*i.e.*, x_N should be ranked higher in the result list). Hence, we use the prediction accuracy to measure the performance of the given mobility model. Given T test movements, for each test movement, we use a mobility model to retrieve the top-k results and see whether the ground truth record x_N appears in the result list. Let T' denote the number of movements for which the ground-truth records are successfully recovered, then the prediction accuracy is $Acc@k = T'/T$. On both LA and NY, we randomly choose 30% movements as test data, and use the rest 70% data for model training.

Compared Methods. To compare with GMOVE, we implemented the following mobility models for context-aware location prediction: (1) **LAW** [2] is a prevailing mobility model. Based on the displacement distribution, it models human mobility as a Lévy flight with long-tailed distributions. (2) **GEO** is based on existing works [13, 4] that train one HMM from the input movements. It uses only spatial information by assuming each latent state generates the location from a bi-variate Gaussian. (3) **SINGLE** is an adaption of GMOVE, which does not perform user grouping, but trains one HMM using all the movements. (4) **NOAUG** is another adaption of GMOVE, which does not perform text augmentation.

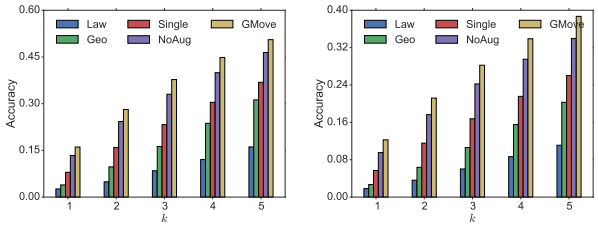
When computing the visiting probability for any candidate location, both GEO and SINGLE simply use the obtained HMM and the forward scoring algorithm to derive the likelihood of the sequence. For GMOVE and NOAUG, they both produce an ensemble of HMMs, and each HMM provides a visiting probability for the candidate location. Hence, GMOVE and NOAUG need to first compute the membership vector for the target user based on her historical movements, and then use membership values as weights to derive a weighted score.

Note that, while there are other methods [14, 15, 5, 19] optimized for location prediction (*e.g.*, by using a supervised learning framework), we have not included them for comparison. This is because GMOVE is designed for mobility modeling, and we use context-aware location prediction as a quantitatively evaluation for the quality of the obtained mobility models. We do not claim GMOVE can outperform state-of-the-art location prediction techniques.

Parameter Settings. GMOVE has four major parameters: (1) the correlation threshold δ; (2) the augmentation size L; (3) the number of user groups G; and (4) the number of latent states K. We study the effects of different parameters as follows: (1) $\delta = 0$, 0.1, **0.2**, 0.3, 0.4, 0.5; (2) $L = 0$, 20, **40**, 60, 80, 100; (3) $G = 5$, 10, 20, 30, 40, 50, 60, 70, **80**, 90, 100; (4) $K = 3$, 4, 5, **10**, 15, 20, 25, 30. When studying the effect of one parameter, we fix the other parameters to their default values, as denoted by the bold numbers.

5.3.2 Prediction Accuracy Comparison.

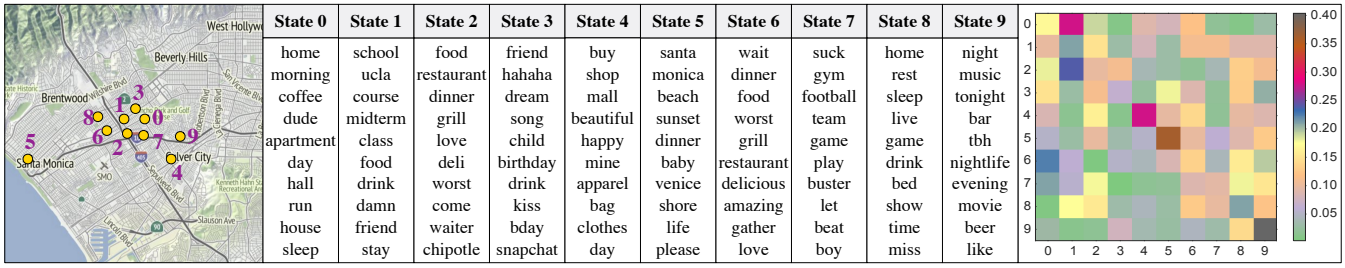
Figure 5 reports the accuracies of different mobility models for top-k prediction on LA and NY. The parameters of GMOVE are set to the default values as such a setting offers the highest prediction accuracy. The parameters of all the other compared models are also carefully tuned to achieve the best performance.



(a) LA (b) NY

Figure 5: Prediction accuracy *v.s.* k.

As shown in Figure 5(a) and 5(b), GMOVE significantly outperforms all the baseline methods on both LA and NY for different k values. Comparing the performance of GMOVE and SINGLE, we find that the prediction accuracy of GMOVE is about 12.7% better on average. This suggests that there indeed exist multiple groups of people that have different moving behaviors. As SINGLE trains one model for all the input movements, it suffers from severe data inconsistency and mixes different mobility regularities together. In contrast, GMOVE employs the iterative refinement framework to perform group-level mobility modeling. Such a framework effectively distinguishes different mobility regularities



State 0	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8	State 9
home	school	food	friend	buy	santa	wait	suck	home	night
morning	ucla	restaurant	hahaha	shop	monica	dinner	gym	rest	music
coffee	course	dinner	dream	mall	beach	food	football	sleep	tonight
dude	midterm	grill	song	beautiful	sunset	worst	team	live	bar
apartment	class	love	child	happy	dinner	grill	game	game	tbh
day	food	deli	birthday	mine	baby	restaurant	play	drink	nightlife
hall	drink	worst	drink	apparel	venice	delicious	buster	bed	evening
run	damn	come	kiss	bag	shore	amazing	let	show	movie
house	friend	waiter	bday	clothes	life	gather	beat	time	beer
sleep	stay	chipotle	snapchat	day	please	love	boy	miss	like

(a) The mobility model for the first user group (students).

State 0	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8	State 9
hotel	restaurant	lax	universal	food	venice	shopping	night	food	club
sleep	food	travel	studio	restaurant	beach	buy	hotel	table	night
miss	amazing	airport	night	chinese	people	shop	live	taste	music
building	eat	international	party	mexican	life	clothes	lounge	salad	girl
center	lunch	come	hollywood	spicy	spa	cheap	place	quiet	nightclub
plaza	grill	flylax	buster	hometown	sunshine	sales	taxi	deli	money
flight	american	arrival	friend	taste	beautiful	nice	shower	hot	drink
home	deli	terminal	happy	china	santa	women	sleep	steak	sleep
city	together	train	haloween	town	monica	mall	queen	dining	dance
inn	pancake	depart	club	delicious	food	men	terminal	fame	rock

(b) The mobility model for the second user group (tourists).

Figure 4: Two group-level mobility models learnt by GMOVE on LA. For each model, we show (1) the geographical center of each latents state; (2) the top eight keywords of each latent state; and (3) the transition probability matrix among the ten states.

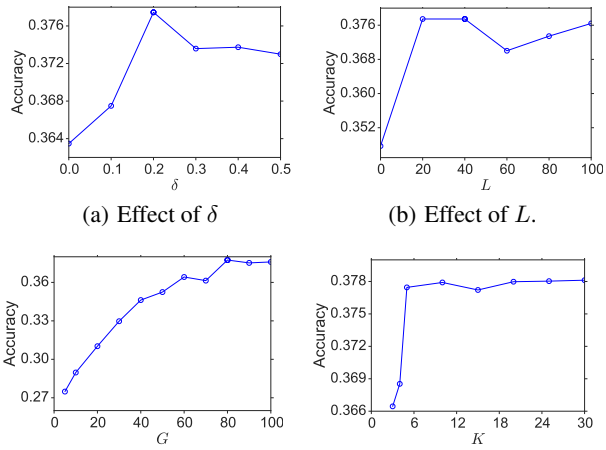
and thus achieves much better accuracy. Meanwhile, by examining the performance of GMOVE and NOAUG, we find that the text augmenter of GMOVE is also effective and improves the prediction accuracy by about 3%. As aforementioned, while each raw tweet message is very short, the augmented message becomes much more semantically comprehensive. The augmentation process allows us to connect the tweets that are intrinsically relevant and thus generate higher-quality HMMs. Another interesting finding is that the performance of SINGLE is consistently better than GEO and LAW. Such a phenomenon verifies the fact that integrating multiple (spatial, temporal, and textual) signals can better describe the users' moving behaviors than using only spatial information, and thus achieve better location prediction accuracy.

5.3.3 *Effects of Parameters.*

As mentioned earlier, there are four major parameters in GMOVE: δ, L, K, and G. After tuning those parameters on LA and NY, we find that the trends for all the four parameters are very similar on LA and NY. We only report the results on LA to save space.

We first study the effects of δ and L. Figure 6(a) shows the prediction accuracy of GMOVE when δ varies from 0 to 0.5. As shown, the performance of GMOVE first increases with δ and then decreases. This phenomenon is expected: a too small correlation threshold tends to include irrelevant keywords in the augmentation process; while a too large threshold constrains a keyword's vicinity to include almost only itself, making the augmentation ineffective. Figure 6(b) shows the performance of GMOVE when the augmentation size L varies. We find that the prediction accuracy first increases with L and quickly becomes stable. This suggest that the augmentation size should not be set to a too small value in practice.

We proceed to study the effects of G and K. Figure 6(c) reports results when G increase from 5 to 100. Not hard to observe, the prediction accuracy increases significantly with G before it gradually becomes stable. This is expected. In big cities like Los Angeles and New York, it is natural that there are a large number of user groups that have totally different lifestyles. When G increases, the GMOVE model fits better with the intrinsic data complexity, and thus improves the prediction accuracy. Figure 6(d) shows the effect



(a) Effect of δ (b) Effect of L.

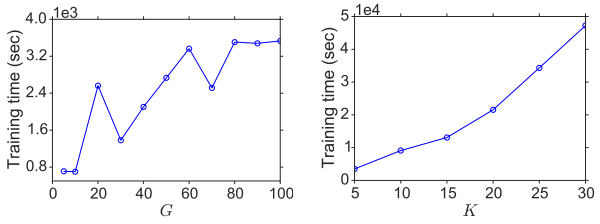
(c) Effect of G. (d) Effect of K.

Figure 6: Effects of parameters (LA, $k = 3$).

of K on the performance of GMOVE. Interestingly, we observe the performance of GMOVE is not sensitive to the number of latent states as long as $K \geq 5$. This suggests that when the user groups are fine-grained enough (*e.g.*, $G = 80$), the number of "hotspots" for each group is usually limited in practice.

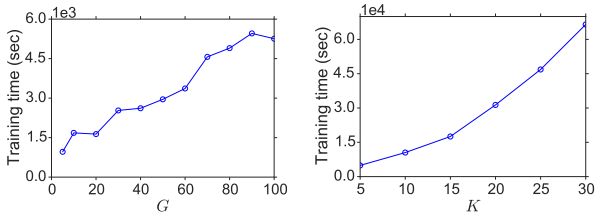
5.4 Efficiency Study

In the final set of experiments, we report the training time of GMOVE. When tuning the four parameters of GMOVE, we find that the training time of GMOVE does not change much with δ and L. Therefore, we only report the effects of G and K. Figure 7(a) and 8(a) show that the time cost of GMOVE generally increases with G, but scales well. This is a good property of GMOVE in situations where G needs to be set to a large number. In Figure 7(b) and 8(b), we can see that the time cost increases superlinearly with K. This is mainly because of the intrinsic nature for HMM training, where time complexity of the EM algorithm is quadratic in K.



(a) Running time *v.s.* G. (b) Running time *v.s.* K.

Figure 7: Running time of GMOVE on LA.



(a) Running time *v.s.* G. (b) Running time *v.s.* K.

Figure 8: Running time of GMOVE on NY.

6. RELATED WORK

Generally, existing approaches on mobility modeling can be classified into three classes: *pattern-based*, *law-based*, and *model-based*.

Pattern-based approaches mine regular mobility patterns from movement data. Existing studies have designed effective methods for mining different types of movement patterns. Specifically, Giannotti *et al.* [6] define a T-pattern as Region-of-Interest (RoI) sequence that frequently appears in the input trajectories. By partitioning the space, they use frequent sequential pattern mining to extract all the T-patterns. Several studies have attempted to find a set of objects that are frequently co-located. Efforts along this line include mining *flock* [10], *swarm* [11], and *gathering* [24] patterns. Meanwhile, there has also been study on mining periodic movement patterns. For instance, Li *et al.* [12] first use density-based clustering to extract reference spots from GPS trajectories, and then detect periodic visiting behaviors at those spots.

While the above studies focus on pure GPS trajectory data, a few studies mine mobility patterns from semantic trajectories. Alvares *et al.* [1] first map the stops in trajectories to semantic places using a background map, and then extract frequent place sequences as sequential patterns. Zhang *et al.* [23] introduce a top-down approach to mine fine-grained movement patterns from semantic trajectories. Our work differs from these studies in two aspects. First, we consider the unstructured GSM text instead of structured category information. Second, instead of extracting a large number of patterns, we develop a statistical model that comprehensively and concisely summarizes people's moving behaviors.

Law-based approaches investigate the physical laws that govern human's moving behaviors. Brockmann *et al.* [2] find that human mobility can be approximated by a continuous-time random-walk model with long-tail distributions. Gonzalez *et al.* [7] study human mobility using mobile phone data. They find that people periodically return to a few previously visited locations, and the mobility can be modeled by a stochastic process centered at a fixed point. Song *et al.* [18] report that around 93% human movements are predictable due to the high regularity of people's moving behaviors. They [17] further propose a self-consistent microscopic model to predict individual mobility.

Model-based approaches attempt to learn statistical models from movement data. Cho *et al.* [3] observe that a user usually moves around several centers (*e.g.*, home and work) at fixed time slots, and thus propose to model a user's movement as a mixture of Gaussians. Their model can further incorporate social influence based on the fact that a user is more likely to visit a location that is close to the locations of her friends. Wang *et al.* [19] propose a hybrid mobility model that uses heterogeneous mobility data for better location prediction. There are some studies [16, 21, 8, 9, 22] on geographical topic discovery, aiming to discover people's activities in different geographical regions and/or time periods. While our work also uses statistical models to unveil human mobility, it differs from the above studies in that it focuses on extracting the latent states of human movements as well as the *sequential transitions*.

The most relevant works to our study are those HMM-based approaches [4, 20]. Mathew *et al.* [13] discretize the space into equal-size triangles using Hierarchical Triangular Mesh. By assuming each latent state has a multinomial distribution over the triangles, they train an HMM on the input GPS trajectories. Deb *et al.* [4] propose the probabilistic latent semantic model, which essentially uses HMM to extract latent semantic locations from cell tower and bluetooth data. Ye *et al.* [20] use HMM to model user check-ins on location-based social networks. By incorporating the category information of places, their obtained HMM is capable of predicting the category of the next location for the user.

Although our proposed GMOVE method also relies on HMM, it differs from the above works in two aspects. First, none of the above studies consider text data for mobility modeling.[3] In contrast, we simultaneously model spatial, temporal, and textual information to obtain interpretable latent states. Second, they perform HMM training with the movement data of either one individual or a given collection of users. GMOVE, however, couples the subtasks of user grouping and mobility modeling, and generates high-quality user groups as well as group-level HMMs.

7. CONCLUSION

In this paper, we studied the novel problem of modeling human mobility using geo-tagged social media (GSM). To effectively unveil mobility regularities from the sparse and complex GSM data, we proposed the GMOVE method. Our method distinguishes itself from existing mobility models in three aspects: (1) it enables the mutual enhancement between the subtasks of user grouping and mobility modeling, so as to generate high-quality user groups and reliable mobility models; (2) it leverages keyword spatiotemporal correlations as auxiliary knowledge to perform text augmentation, which effectively reduces text sparsity present in most GSM data; and (3) it integrates multiple (spatial, temporal, and textual) signals to generate a comprehensive view of users' moving behaviors. Our experiments on two real-life data sets show that the strategies employed by GMOVE are highly effective in generating quality mobility models. Meanwhile, using context-aware location prediction as an example task, we found GMOVE can achieve much better prediction accuracy than baseline mobility models.

There are several future directions based on this work. First, we currently use a pre-specified time constraint to extract reliable movements from raw GSM history. In the future, it is interesting to adapt GMOVE such that it can automatically infer the reliability of the transitions and extract movements from the raw data. Second, besides context-aware location prediction, we plan to apply GMOVE for other tasks in the urban computing context. By the year 2030, it is estimated over 60% of the world population will be

[3] Although [20] considers categories of the places, such information is typically unavailable in GSM data, and extracting semantics from unstructured text is more challenging.

located within a city's boundary. One of the huge challenges facing many government organizations, including the US military, is understanding and preparing for complex urban environments. From disaster relief to anomaly movement detection, the challenges associated with operating within the so-called mega-cities are many. The ability to accurately model group-level human mobility will dramatically improve the understanding of normal urban life patterns and help address such challenges.

8. REFERENCES

[1] L. O. Alvares, V. Bogorny, B. Kuijpers, B. Moelans, J. A. Fern, E. D. Macedo, and A. T. Palma. Towards semantic trajectory knowledge discovery. *Data Mining and Knowledge Discovery*, 2007.

[2] D. Brockmann, L. Hufnagel, and T. Geisel. The scaling laws of human travel. *Nature*, 439(7075):462–465, 2006.

[3] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: User movement in location-based social networks. In *KDD*, pages 1082–1090, 2011.

[4] B. Deb and P. Basu. Discovering latent semantic structure in human mobility traces. In *Wireless Sensor Networks*, pages 84–103. Springer, 2015.

[5] H. Gao, J. Tang, and H. Liu. Mobile location prediction in spatio-temporal context. In *Nokia mobile data challenge workshop*, volume 41, page 44, 2012.

[6] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. Trajectory pattern mining. In *KDD*, pages 330–339, 2007.

[7] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.

[8] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, and K. Tsioutsiouliklis. Discovering geographical topics in the twitter stream. In *WWW*, pages 769–778, 2012.

[9] B. Hu and M. Ester. Spatial topic modeling in online social media for location recommendation. In *RecSys*, pages 25–32, 2013.

[10] P. Laube and S. Imfeld. Analyzing relative motion within groups of trackable moving point objects. In *GIScience*, pages 132–144, 2002.

[11] Z. Li, B. Ding, J. Han, and R. Kays. Swarm: Mining relaxed temporal moving object clusters. *PVLDB*, 3(1):723–734, 2010.

[12] Z. Li, B. Ding, J. Han, R. Kays, and P. Nye. Mining periodic behaviors for moving objects. In *KDD*, pages 1099–1108, 2010.

[13] W. Mathew, R. Raposo, and B. Martins. Predicting future locations with hidden markov models. In *UbiComp*, pages 911–918, 2012.

[14] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. Wherenext: a location predictor on trajectory pattern mining. In *KDD*, pages 637–646, 2009.

[15] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo. Mining user mobility features for next place prediction in location-based services. In *ICDM*, pages 1038–1043, 2012.

[16] S. Sizov. Geofolk: latent spatial semantics in web 2.0 social media. In *WSDM*, pages 281–290, 2010.

[17] C. Song, T. Koren, P. Wang, and A.-L. Barabási. Modelling the scaling properties of human mobility. *Nature Physics*, 6(10):818–823, 2010.

[18] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[19] Y. Wang, N. J. Yuan, D. Lian, L. Xu, X. Xie, E. Chen, and Y. Rui. Regularity and conformity: Location prediction using heterogeneous mobility data. In *KDD*, pages 1275–1284, 2015.

[20] J. Ye, Z. Zhu, and H. Cheng. What's your next move: User activity prediction in location-based social networks. In *SDM*, 2013.

[21] Z. Yin, L. Cao, J. Han, J. Luo, and T. S. Huang. Diversified trajectory pattern ranking in geo-tagged social media. In *SDM*, pages 980–991, 2011.

[22] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Who, where, when and what: discover spatio-temporal topics for twitter users. In *KDD*, pages 605–613, 2013.

[23] C. Zhang, J. Han, L. Shou, J. Lu, and T. F. L. Porta. Splitter: Mining fine-grained sequential patterns in semantic trajectories. *PVLDB*, 7(9):769–780, 2014.

[24] K. Zheng, Y. Zheng, N. J. Yuan, and S. Shang. On discovery of gathering patterns from trajectories. In *ICDE*, pages 242–253, 2013.

[25] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from gps trajectories. In *WWW*, pages 791–800, 2009.

APPENDIX

We derive the updating rules for the HMM parameters in the M-step as follows. Given the weighted movements $\{S_1, S_2, \ldots, S_R\}$ and the old paramter set $\Phi^{(t)}$, the expectation of the complete likelihood is

$$Q(\Phi) = \sum_{r=1}^{R} \sum_{z_r} w_r p(z_r|S_r;\Phi^{(t)}) \ln p(S_r, z_r|\Phi) \tag{14}$$

$$= \sum_{r=1}^{R} \sum_{z_r} w_r p(z_r|S_r;\Phi^{(t)}) \ln p(z_{r,1}|\pi) \tag{15}$$

$$+ \sum_{r=1}^{R} \sum_{z_r} \sum_{n=2}^{N} w_r p(z_r|S_r;\Phi^{(t)}) \ln p(z_{r,n}|z_{r,n-1}, \mathbf{A}) \tag{16}$$

$$+ \sum_{r=1}^{R} \sum_{z_r} \sum_{n=1}^{N} \sum_{k=1}^{K} w_r p(z_r|S_r;\Phi^{(t)}) z_{r,n}^k \ln p(x_{r,n}|\Phi). \tag{17}$$

Meanwhile, since $z_{r,n}^k$ is a binary variable, we have:

$$\gamma(z_{r,n}^k) = \sum_{z^r} p(z^r|S_r;\Phi^{(t)}) z_{r,n}^k \tag{18}$$

$$\xi(z_{r,n-1}^j, z_{r,n}^k) = \sum_{z^r} p(z^r|S_r;\Phi^{(t)}) z_{r,n-1}^j z_{r,n}^k \tag{19}$$

The three terms in Equation 15, 16, and 17 are independent, we can thus optimize the parameters π, $\mathbf{A}$, and $\{\mu_l, \Sigma_l, \mu_t, \sigma_t, \theta\}$ separately. To optimize π, we need to maximize Equation 15, namely

$$f(\pi) = \sum_{r=1}^{R} \sum_{z_r} w_r p(z_r|S_r;\Phi^{(t)}) \sum_{k=1}^{K} z_{r,1}^k \ln \pi_k \tag{20}$$

$$= \sum_{r=1}^{R} \sum_{k=1}^{K} w_r \gamma(z_{r,1}^k) \ln \pi_k. \tag{21}$$

Using the Lagrange multiplier, we can easily obtain the updating rule for π (Equation 1) that maximizes the above objective function.

Now consider the optimization of A. With Equation 16 and 19, we can write down the following objective function, which can be similarly optimized using use the Lagrange multiplier to produce Equation 2:

$$g(A) = \sum_{r=1}^{R} \sum_{n=2}^{N} \sum_{k=1}^{K} \sum_{j=1}^{K} \xi(z_{r,n-1}^j, z_{r,n}^k) \ln A_{jk} \tag{22}$$

Finally, as the spatial, temporal, and textual observations are independent, the parameter sets $\{\mu_{kg}, \Sigma_{kg}\}$, $\{\mu_{kt}, \Sigma_{kt}\}$, and θ_k can be derived separately based on 17. Recall that the spatial and temporal distributions are assumed to be Gaussian, we simply take the derivatives of the objective function and set them to zeros, which produces the updating rules in Equation 3, 4, 5, and 6. To optimize θ_k, we again combine Equation 18 with the Lagrange multiplier, and obtain the maximizer shown in Equation 7.