Enhanced Group-Level Mobility Modeling Using Text Embedding

Xiaobin Gao (xgao16), Wanzheng Zhu (wz6)*

May 9, 2017

Abstract

Understanding and modeling human’s mobility plays a central role in various areas. Because of its great importance, the problem has been extensively studied from different angles. Recently, the rapid development of geo-tagged social media has brought both opportunities and challenges to mobility modeling. In particular, a state-of-the-art work named ”GMove” interests us. GMove uses Hidden Markov Model to capture human’s mobility. The observation state is a triplet consisting of location, time stamp, and a bag of keywords. Although GMove has achieved remarkable success, yet we think there is still room for improvement. What motivate us are the novel works on text embedding, that is, word2vec and sentence2vec, which have been demonstrated to have strong power in overcoming the drawbacks of bag-of-words model. In this work, we apply the Von Mises-Fisher distribution to bridge sentence2vec with GMove. We also conduct experiments to evaluate the performance of our model. Experimental results show that our model has a similar accuracy with GMove in predicting the next step movement of a specific user. Interestingly, it requires much shorter training time.

1 Introduction

1.1 Background

Understanding human’s mobility is extremely important and has various applications. For example, if one can model the traffic flow, then the model can be applied to predict when and where there might be a traffic jam. This prediction can help the government do a better traffic control such that the potential traffic jam can be reduced or even avoided. As another example, if the shop like Macy’s can build a model predicting the movement of a person, say he or she is likely to go shopping in the next few days, then the seller can send some advertisements to that person. Due to its great importance, this problem has been extensively studied, see [1–15]. These mentioned works take GPS data as input. Although the data can be obtained from the GPS units embedded in the devices...
like cell phones, yet obtaining GPS data for research and development can be very expensive (need to buy the data from cell phone carriers) or even not permitted (due to privacy issue). In recent years, social media, especially geo-tagged social media like twitter or Facebook check-in, have grown rapidly. This rapid development introduces a new angle to access the mobility modeling problem. Compared with GPS data, geo-tagged social media data has the following properties: first, it is much easier to obtain and thus the size of the data set is considerably large. Second, it is sparse, as people will not keep posting tweets all the day. Third, it contains text data, which conveys rich semantic meanings. Therefore, geo-tagged social media bring us both opportunities and challenges.

1.2 Motivation

One of the state-of-art results taking advantage of geo-tagged social media in mobility modeling has been discussed in [16], which is called “GMove”. In GMove, mobility is captured by a Hidden Markov Model (HMM). The observation state of the HMM is a triplet consisting of the location state (“where”), the time state (“when”), and the behavior state (“what”). Specifically, the behavior state is described by a bag of keywords, which is obtained from the tweet using a sampling-based method. Furthermore, the generation from the latent state to the behavior state is modeled as multinomial distribution.

Although GMove has achieved a remarkable success, yet there is still room for improvement. What attracts us is the bag-of-words model used to capture the behavioral information. It is well known that bag-of-words model has two major drawbacks: first, it does not contain information about the order of words, which results in loss of semantic meanings. Second, it is high-dimensional and requires training on many parameters. Consequently, the training time of the bag-of-words model is considerably long. Recent works [17–21] have shown that text embedding techniques, e.g., word2vec, sentence2vec, and LINE, are very effective in dealing with the two drawbacks discussed above. Hence, it motivates us to apply text embedding technique to enhance the performance of GMove.

1.3 Contributions

In this work, we apply text embedding techniques to enhance the method described in GMove. First, we do an experiment (similarity check) to demonstrate the power of sentence2vec in capturing the semantic meanings of tweets. Then, we modified the HMM by replacing the behavior state described by bag-of-words with that described by a low-dimensional vector. Correspondingly, the generative relationship from the latent state to the observation state needs to be changed. Note that two vector representations carry similar semantic meanings if they have the similar direction or small cosine distance. This phenomenon shows that the semantic meaning carried by a vector strongly depends on its direction. Hence, we need to select a probability distribution that can model the generation of vectors with different angles. After a careful search, we choose Von-Mises Fisher (VMF) distribution, which is a distribution over the surface of a multi-dimensional sphere. Next, we re-derive the Expectation-Maximization (EM) update rule for estimating parameters of HMM. Due to the complicated probabilistic description of VMF, it is difficult to derive a closed-form expression
for the update rule. Consequently, we derive an approximate update rule for EM, and it shows that this iteration rule converges during the experiment. Then, we conduct experiments on predicting individual’s next step of movement. Experimental results show that our model does not outperform GMove. To analyze the insights of the results, we do some auxiliary experiments. Surprisingly, when we remove the text information from the tweets, the performances of our system and GMove decrease only a little bit, which indicates that this specific prediction task may not strongly rely on the text information. We also analyze and discuss some other underlying factors responsible for the experimental results. On the other hand, although our model does not outperform GMove in accuracy, yet it has much shorter training time. Finally, we come up with interesting future works including evaluation of our model under some other tasks, which may better take advantage of text information. Our major contributions are listed as follows:

1. We conduct experiments to demonstrate the power of sentence2vec in capturing the semantic meanings of tweets, which supports our motivation.

2. We conduct extensive literature review to select the proper probability distribution (VMF) modeling the generation of vector representations.

3. We derive an approximate EM iteration rule under the VMF distribution, as the exact update rule does not have a closed-form expression. It shows that the approximate iteration rule converges when we conduct experiments.

4. We conduct experiments to evaluate the performance of our model. We also do auxiliary experiments to help analyze the results.

5. We carefully analyze and discuss factors leading to the experimental results, which also serve as guidance of future research directions.

1.4 Organization

The rest of the report is organized as follows: in Section 2, we will present GMove in detail, since it serves as base of our work. In Section 3, we will introduce sentence2vec and explain how it works. In Section 4, we will present details about VMF distribution, which include an introduction on this distribution (“what”), the reason we choose this distribution (“why”), and the re-derived EM iteration rule (“how”). In Section 5, we will present and analyze experimental results. We first present a demo illustrating the power of sentence2vec in capturing semantic meanings of tweets. Then, we present experimental results evaluating the accuracy of GMove and our model in predicting the next step movement of a person. Next, we compare the training time of GMove and our model under the various data sets. In Section 6, we draw concluding remarks and discuss future directions for research.
2 A Recap of GMove

Since our work is built on GMove [16], we would like to introduce GMove first. In this section, we will focus on answering the following questions:

1. Why does GMove build group-level model instead of personalized model?
2. How does GMove model human’s mobility?

2.1 Group-Level Mobility Modeling

Although geo-tagged data is considerably large in size, yet it is very sparse. That is, the data from each person is quite limited. Hence, we do not have enough data to train a personalized model for each person. In order to deal with data sparsity, we need to aggregate the data together. Then, the question is whether we should aggregate all the data together to build one model working for everyone. The problem with this idea is that when aggregating all data together, the data becomes very inconsistent. This is quite intuitive as people’s movement behavior is really diverse. To aggregate data together without compromising the consistency, GMove proposes an idea that we can assign users into different groups such that the users in the same group share similar movement regularity. Then, the data in each group is consistent and substantial, and we will be able to build one mobility model for each group.

Now there raises another task other than mobility modeling, that is, user grouping task. GMove further discusses that these two tasks can mutually enhance each other. On the one hand, if we properly assign user into groups, the geo-tagged data over each group will be very consistent, and thus we may build good mobility models. On the other hand, good mobility models also provide quality supervision in user grouping. For example, if we have a quality mobility model for one user group, when a user comes, we have confidence to determine whether the user belongs to this group or not. To be more specific, we simply needs to compare the movement behavior of the user with the movement behavior described by the group mobility model. If the behaviors are close to each other, then with high probability the user belongs to the group. The confidence of the judgement comes from the quality of the model. Hence, good mobility modeling does enhance user grouping.

2.2 Hidden Markov Model for Mobility Modeling

GMove uses Hidden Markov Model (HMM) to capture group mobility. Note that using HMM for mobility modeling is also a popular approach, which has shown in [4, 22, 23]. The Hidden Markov Model can be described by a graph, as shown in Fig.1. In the graph, the circles represent states, the directed link represent the dependency relationship. For example, $Z_1$ points to $Z_2$ means “$Z_1$ imply $Z_2$”, or “$Z_2$ depends on $Z_1$”. To help understanding, one may think of this relationship as “$Z_2$ is evolved from $Z_1$, and thus it depends on $Z_1$”. Similarly, one can see that $X_1$ depends on $Z_1$. The upper level states, or $Z_1, Z_2, \ldots, Z_N$, are called the latent states. The latent states together with their interrelationship forms a Markov chain. In general, latent states may or may not be
interactable. However, here we may explain them as what’s happening in real life. For example, 
$Z_1$ can be what a user is doing in the morning, $Z_2$ can be what the user is doing in the afternoon, 
etc. Note that $Z_1, Z_2, \ldots, Z_N$ are random variables as what a person is doing in the morning can 
very quite a lot. The evolution between states $Z_1$ and $Z_2$ is described by the probability transition 
matrix, which is a parameter of the model. The lower level states, i.e., $X_1, X_2, \ldots, X_N$, are called 
the observation states. The observation states are what we really observe. For instance, we cannot 
access a user’s room and observe what the user is doing. Instead, we can only see the tweets posted 
by the user. Hence, the observation state is something we can obtained from the tweets. Since 
the tweets posted by a user is a capture of what the user is actually doing, we say the observation 
states are “generated” from the latent states and the relationship between the latent states and the 
observation states is called generative relationship. Note that the generative relationship is always 
described by a probability distribution and consequently, the observation states are also random 
variables. To understand the reason, one may think about a case that even though a user is having 
breakfast at home, he or she may post a tweet like “Nice breakfast” or “Sun shines in the morning”. 
To model the diversity of possible outcomes, we use probability distribution. What a probability 
distribution can tell us is that with high probability, a user in the morning may post the tweet 
related to breakfast but not lunch. From the description above, one can see that Hidden Markov 
Model easily matches the context of mobility modeling using geo-tagged data.

In GMove, the observation state $X_n$ is a triplet containing the time state $t_n$ (time stamp), the 
location state $l_n$ (longitude and latitude), and the behavior state $e_n$. The time and location states 
can be easily obtained from the geo-tagged tweet. In particular, the behavior state $e_n$ is a bag of 
keywords. The keywords of a tweet are sampled from a candidate pool containing words from that 
tweet and some neighboring words. The neighboring words are some words whose spatialtemporal 
distribution of occurrence is similar to that of the centering word. The purpose of sampling a bag 
of keywords is to argument the semantic meanings, as otherwise there might be just a few keywords 
reflecting weak semantic meanings.

We think the framework of GMove is really nice, yet the bag-of-words model used in HMM can 
be further improved. To be more specific, we would like to replace it with a lower dimensional 
vector generated from sentence2vec (which will be discussed in Section 3). Correspondingly, the 
current generative relationship from latent state to the bag of words is changed from multinominal 
distribution to Von Mises-Fisher distribution (which will be discussed in Section 4). Before moving 
on, we would like to present Fig.2, which clearly shows the difference between HMM used in GMove
Although the bag-of-words model is very popular and easy to implement, yet it suffers from two major drawbacks: first, bag-of-words model does not take the word order information into account, and thus it may miss capturing some semantic meanings. Second, due to the high-dimensional nature of text, the training time of bag-of-time model is usually long. Recent works on text embedding (see [17–21]) have achieved great success in dealing with the drawbacks of bag-of-words model. Here, we will discuss two models, namely the word2vec model (proposed in [17, 18]) and the sentence2vec model (proposed in [19]), whose frameworks are described in Fig. 3.

The idea is word2vec is as follows: good vector representations for words should do a good job in a prediction task. To be more specific, suppose we are given the vector representations of context words \( w_{t-k}, \ldots, w_{t+k} \), we can compute the likelihood of each candidate center words, i.e.,

\[
p(w_t|w_{t-k}, \ldots, w_{t+k}).
\]

To predict which candidate word is the center word, we simply select the one achieving the maximum likelihood. If the word embeddings are effective, then the likelihood of the real center word should be high. Mathematically, the vectors are selected to maximize the likelihood of the whole corpus,
that is,
\[
\frac{1}{T} \sum_{t=k}^{T-k} \log p(w_t|w_{t-k}, \ldots, w_{t+k}),
\]
where \( T \) is number of words in the corpus. Similar to word2vec, sentence2vec simply modifies to likelihood function as
\[
p(w_t|w_{t-k}, \ldots, w_{t+k}, d_m),
\]
where \( w_{t-k}, \ldots, w_t, \ldots, w_{t+k} \) belongs to paragraph \( m \) (which can also be a sentence), whose vector representation is \( d_m \). Again, if the word embedding and the paragraph (sentence) embedding are effective, the likelihood of the “true” center word will be high. By applying the sentence2vec model, we can embed each tweet into a low-dimensional vector. In Section 5, we will present qualitative experimental results showing the power of sentence2vec in capturing the semantic meanings of tweets.

4 Von-Mises Fisher (VMF) Distribution

In the previous sections, we talked about “GMove” has used bag-of-words to represent text information. In order to model bag-of-words into the HMM model, we adopt a multinomial distribution, i.e. \( p(e_n|z_n = k) \propto \prod_{v=1}^{V} \theta^v_{k_v} \), where \( e \) indicates how many times \( v \) has occurred in \( e_n \), and \( \theta \) indicates the probability of word \( v \) chosen at state \( k \).

However, the same distribution is no longer valid for continuous sentence2vec vectors, because: (1) vectors generated from sentence2vec are continuous-valued vectors; and (2) there is no occurred indicator for each value inside the vector. Therefore, we must need to find another mathematical distribution to represent sentence2vec vectors and fit the text vectors into this model.

In the next few subsections, we will what Von-Mises Fisher (VMF) distribution is and then discuss the intuition of why we select Von-Mises Fisher (VMF) distribution [24], and how to update the parameters in the VMF distribution.
4.1 Von-Mises Fisher (VMF) distribution

Von Mises-Fisher distribution denotes a probability distribution on a \((p - 1)\)-dimensional sphere in \(\mathbb{R}^p\), in directional statistics. When \(p = 2\), the distribution becomes von Mises distribution on the circle.

The VMF distribution requires 2 parameters, namely \(\mu\) and \(\kappa\), which denote the mean direction and concentration degree respectively. For a given \(\mu\) and \(\kappa\), the probability density function for a unit vector \(x\) is given by

\[
 f_p(x; \mu, \kappa) = C_p(\kappa) \exp(\kappa \mu^T x)
\]

where \(\mu\) is the mean direction and the 2nd-norm of \(\mu\) must be equal to 1. \(\kappa\) is called the concentration parameter, and it must be of a non-negative value. \(x\) is a unit vector. \(p\) denotes the dimensionality of the vector. \(C_p(\kappa)\) is called the normalization constant, and it is given by

\[
 C_p(\kappa) = \frac{\kappa^{p/2 - 1}}{(2\pi)^{p/2} I_{p/2 - 1}(\kappa)}
\]

where \(p\) is the dimensionality of the vector, and \(I_\alpha\) is the modified Bessel function of the first kind at the order \(\alpha\), and \(I_\alpha(x)\) is given by solving the following differential equation

\[
 x^2 \frac{d^2 y}{dx^2} + x \frac{dy}{dx} + (x^2 - \alpha^2) y = 0
\]

The modified Bessel function denotes the solution for a special case, which is for the purely imaginary argument. The explicit mathematical formula is given as follows:

\[
 I_\alpha(x) = i^{-\alpha} J_\alpha(ix) = \sum_{m=0}^{\infty} \frac{1}{m! \Gamma(m + \alpha + 1)} \left( \frac{x^2}{2} \right)^{2m+\alpha}
\]

where \(\Gamma(m + \alpha + 1)\) denotes the gamma function for non-integer values.

A special case for VMF distribution is when \(\alpha\) is 3, then the normalization constant becomes

\[
 C_3(\kappa) = \frac{\kappa}{4\pi \sinh \kappa} = \frac{\kappa}{2\pi (e^\kappa - e^{-\kappa})}
\]

This is as known as Fisher distribution, which is originally used to model the interaction of electric dipoles in an electric field. VMF distribution is an extremely math-complicated distribution. However, we can qualitatively analysis this distribution. Since \(x\) is a \(p\)-dimensional vector, the parameter \(\mu\) denotes the the mean direction of the distribution. The parameter \(\kappa\) is a concentration parameter, which denotes the concentration around the mean direction \(\mu\). Qualitatively, the larger the concentration parameter \(\kappa\) is, the denser the vectors are around the mean direction. If \(\kappa\) is 0, then it can be considered as a uniform distribution over \(p\)-dimensional vectors.

To provide a more intuitive understanding about VMF distribution, we present what a typical 3-dimensional VMF looks like and it is shown in Fig. 4.

As we can see, there are three different distributions (different mean direction \(\mu\) and different concentration parameter \(\kappa\)). For each distribution, we randomly sample 1000 points and present
them in this 3D sphere. The mean directions $\mu$ are denoted by the arrow (red, green, blue). The concentration parameter $\kappa$ is 1 for blue, 10 for green and 100 for red. The larger the concentration parameter, the more samples are towards the mean direction.

4.2 Intuition of adopting VMF distribution to model text data

In the previous subsection, we have analyzed the VMF distribution quantitatively and qualitatively. In this subsection, we will talk about our intuition of why we would like to model the sentence2vec vectors in the VMF distribution.

The two main intuitions are:

1. Sentence2vec can best capture the similarities between sentences and therefore, sentences with similar semantic meanings should appear near in the $p$-dimensional space (In order words, their dot product of sentences with similar semantic meanings should be close to 1);

2. For each observed state in the HMM model, we should expect tweets text has a concentrated topic and thus, for each hidden state, we should expect to have a concentrated series of vectors.

For example, at the state of “6pm” in the HMM model, the group “tourists” will probably send tweets about dinner, food and restaurants. Therefore, for the vectors in VMF distribution, we should expect to see a dense distribution around the mean direction of “food” and “restaurants”. We should expect to see fewer tweets about “coffee”, “tea” and “dessert”. We should expect to see very fewer tweets about “exams”, “anxiety” and “murderer”. Since sentence2vec vectors can capture the semantic meanings of each tweet, it corresponds to a dense distribution to vector “food” and vector “restaurant” and a sparse distribution to other vectors.
Based on the above intuition, we find that VMF distribution can capture a reasonably good semantic topic center of tweets, and therefore, is chosen to fit the tweets’ text vectors.

4.3 EM on the VMF distribution

To use EM algorithm to update HMM parameters, we need an update rule for VMF parameters. To the best of our knowledge, due to the intrinsic mathematical-complicated form of VMF distribution, there is no close-form solution for the parameters’ update \[26, 27\]. The state-of-the-art approximation for parameters’ update in EM algorithm is proposed by A. Banerjee, etc. \[28\]. Based on their work, we derived the VMF parameters’ approximation update rule:

\[
\mu_{\text{new}} \leftarrow \sum_{r=1}^{R} \sum_{n=1}^{N} \omega_r \gamma(z_{r,n}^k) e_{r,n} \\
\bar{r} = \frac{||\mu||}{N} \\
\mu_{\text{new}} = \frac{\mu}{||\mu||} \\
\kappa_{\text{new}} = \bar{r} d - \bar{r}^3 \\
\text{where } d = \frac{1}{1 - \bar{r}^2}
\]

In our experiment, we successfully achieve the convergence of EM algorithm by using the above update rule.

5 Experiments and Results

In this section, we will evaluate our proposed method, compare our results with the current state-of-the-art method and analyze the results.

5.1 Data sets and pre-processing

Our raw data set contains around 800,000 geo-tagged tweets, which were posted in the Los Angeles area during Aug 1, 2014 to Oct 30, 2014. Each geo-tagged tweet contains a tweet id, a user id, time, location (longitude and latitude), and the text. We define a trajectory as a set of two tweets: one starting tweet and one ending tweet. The two tweets should be posted by the same user sequentially. Furthermore, the time gap between two tweets should be no less than 1 hour and no longer than 3 hours. Under this rule, we extract around 30,000 trajectories, which form the clean data set.
5.2 Similarity Check Using Sentence2vec

To verify that sentence2vec can capture the semantic meanings of tweets, we conduct experiments on the task of similarity check. We first apply sentence2vec to embed all the tweets (∼800,000 in total) into 100-dimensional vectors. Then, given a query tweet, we find the top-100 nearest (smallest cosine distance between vector representations) tweets, and the results are as follows.

Query tweet: “I’m at Hana Sushi Bar in Northridge, CA”.
Neighbor tweet: “I’m at Chilis Grill & Bar in Northridge CA”.
Neighbor tweet: “I’m at SanSai Japanese Grill in Burbank CA”.

We select two representative tweets neighboring to the query tweet. Clearly, both tweets have similar semantic meanings with the query tweet. The first tweet share many words with the query tweet, and thus it should not be hard to find this tweet even if we use the bag-of-words model. The second tweet, however, has little overlap with the query tweet, which demonstrates the power of sentence2vec.

One may question that the example described above is not representative, as both the query tweet and the neighbor tweets are clean. We also conduct the similarity check on noisy query tweet. It turns out that if the query tweet is not too noisy (understandable), then sentence2vec will be able to find reasonable neighbor tweets.

5.3 Quantitative Evaluation

In this subsection, we will explain how we evaluate our model quantitatively and compare results between our proposed model and the current state-of-the-art method.

5.3.1 Evaluation method

Our main goal is to predict a user’s next move given his/her previous moves. Formally speaking, given the ground truth of a user’s trajectory $= x_1, x_2, \ldots, x_N$, we would like to predict $x_N$ in a candidate pool, based on the information of $x_1, x_2, \ldots, x_{N-1}$. The candidate pool is generated as follow: From all the tweets that we have, we select the tweets that has a geographical distance less than or equal to 3 kilometers from $x_N$, and has a time difference less than or equal to 300 seconds. For each prediction task, we firstly generate the candidate pool for this user and then use the above HMM model to predict which is the most likely observation state for this user’s next move. Our prediction is simply calculated as number of correct predictions and number of total predictions. We will also compare top-$k$ predictions with $k = 2, 3, 4, 5$.

The above evaluation method is set up based on the intuition that (1) it is almost impossible to predict exactly what a user’s next move is; and (2) the better the model is, the likely we can select $x_N$ from the candidate pool. Therefore, we will mainly compare prediction results between different models. We calculate prediction accuracy on top 1 to top 5 prediction results.
5.3.2 Results Comparison between proposed model and GMove

Figure 5: Results comparison between Bag-of-words and sentence2vec. Orange columns indicate prediction accuracy with bag-of-words text embedding information and grey columns indicate prediction accuracy with sentence2vec vector embedding information.

Fig. 5 shows the results from the previous GMove model and our proposed model. The leftmost one denotes the top 1 prediction accuracy, and followed by top 2 prediction accuracy and so on. The rightmost one denotes the top 5 prediction accuracy.

From the results we had, we did not achieve a better result than the previous state-of-the-art model. However, we have actually solved some intrinsic problems with bag-of-words and a big improvement should be expected. But why isn’t there an improvement of accuracy? We will address this problem in the next few subsections.

5.3.3 Results comparison with no text information

In our experiment, we have an underlying hypothesis that text information plays an important role in this task. To verify if this hypothesis makes sense, we designed another experiment. We maintain all the information of temporal information and location information, but we deleted all the text information and use vector ”1” for all tweets and see how the results go. The results are summarized in Fig.6.

As we can see from the above figure, it turns out that even without any text information, we can still achieve a reasonable good result. In other words, text information does not provide too much useful information in GMove and our proposed model. Therefore, there are two possibilities. One possibility is that text information does not actually play an important role in this particular task. In other words, in this task, there are much more important factors other than text information. The other possibility is that both of our models do not capture the text information well. The above two reasons will be further discussed in detail in the next two subsections.
5.3.4 Effectiveness of the task

In this subsection, we would like to discuss how important the text information’s role is for this particular task.

First of all, our prediction task is to predict a single user’s next tweet, given his/her previous tweets. Therefore, our main task is to predict human’s behavior in the sense of user-level. However, our model is built based on a group-level.

We have enough reasons to believe that our proposed model and GMove model are able to predict what the most likely move for a group of people, such as tourists, students and professors is. For example, given that students have just finished their lectures and it is about the noon time, it is intuitively reasonable to predict that they are going for lunch. However, for a specific student, there is too much variance. He could have a paper deadline today, and he could have a presentation tomorrow and therefore, he might not choose to go to the restaurant for lunch but instead, he might choose to spend more time on his work. Therefore, for a specific user, it is almost impossible to predict his behavior accurately.

Given that previous state-of-the-art model and our proposed model are not perfectly designed for this task, it is reasonable that experiment results did not outperform the one with no text information.

Besides that, it is also possible that temporal information and location information play a more important role than the text information. The intuition is as follows: a student will probably go home after 10 pm on weekdays. No matter what text tweets he sends, he would probably send tweets at his home, which has the same location. Therefore, location clearly weigh more than the text information. This could be a potential reason that we text information does not play an
important role.

### 5.3.5 Effectiveness of capturing the text information

The previous subsection provides a possibility that trivial improvement over no text tweets can be due to the ineffectiveness of the ability to capture the text information.

As we have discussed before, bag-of-words have some intrinsic problems such as it loses the word order and thus give bad semantic meaning of a sentence. For sentence2vec, we indeed can achieve a much better representation of the text information. Yet, we might introduce more noise. Tweets are considered as one of the noisiest text data in the word. For example, when people send tweets of a “no”, many people will choose to send “NOOOOOOOOO!” with a lot of O’s inside. Besides it, people will make new words such as ”gooooooooood” instead of “good”, “btw” instead of “by the way”, “tc” instead of “take care”. Given so many noisy text, sentence2vec may also learn something from these noisy texts, which may provide some bad text representations.

Also, when we did sentence2vec embedding, we use all the tweets’ text for the embedding process. However, in the experiment, we did a lot of preprocessing such as selecting trajectories and remove advertisement tweets and so on. We end up use only a small portion of the tweets for training and testing. Therefore, the sentence2vec vectors learnt from all raw tweets may not be the best for modeling text information. One interesting thought is to try to do embedding on only the selected tweets.

### 5.3.6 Suggestions of new tasks

For the previous task, we listed a few reasons that why our model did not outperform the previous model. In this subsection, we would like to propose some new tasks on which our proposed model is likely to perform well.

Since our proposed model and GMove are both group-level based models, we can consider to perform some group-level tasks. For example, given all the tweets in Chicago, we can use our model to identify who are from the group of tourist and then outline a one-day Chicago tour. In this task, we only consider a group-level movement and therefore, our model is likely to perform well.

Alternatively, our proposed model is able to capture the text information better and thus is good at perform text related tasks. For example, if we have all the tweets information of UIUC and we would like to distinguish the group of students and the group of professors. The intuition is that professors and students would tend to go to similar locations at similar time but do different things. For example, they will come to the school in the morning and go to have lectures, and then go to their offices and do their work. The locations and time information are almost the same for professors and students. However, the text information should be quite different. Students might be more interested in sends tweets about lectures in the morning and afternoon, while professors might be more interested in papers and meetings.
5.3.7 Computational Complexity

All our above subsections focus on comparing the accuracy of our model and the GMove model. In this subsection, we will discuss another important role to evaluate, namely computational complexity.

We ran all the experiment on a MacBook Pro with a 2.9 GHz Intel Core i7 CPU, a 16 GB 2133 MHz memory and macOS Sierra 10.12.4 operating system.

![Timing Analysis](image)

**Figure 7:** Time complexity comparison between bag-of-words and Sentence2vec. The presented value are the averaged time in seconds.

We did experiments over different number of latent states, different number of candidate pool size and different number of user groups. The average running time for bag-of-words is 282 seconds, and the average running time for Sentence2vec is 117.2 seconds, as shown in Fig.7. Our proposed method achieves a 60% less time for training and testing.

The main reason is that bag-of-words in GMove have too many parameters to be updated. GMove adopts a strategy called text augmenter and it will assign probability for all the keywords. Therefore, for each hidden state of the HMM model, for each keyword, there is a probability value to be updated. There are more than 7500 keywords in our experiments. However, for sentence2vec in VMF distribution, at each state, there are only two parameters requiring updating, mean direction and the concentration parameter. Although mean direction is a 100-dimensional vector, it still requires a lot lesser parameters’ update.

6 Conclusion

In this work, we have tried to apply the embedding technique, sentence2vec, to enhance the group-level mobility modeling method, GMove, proposed in [16]. We have done experiments to verify that sentence2vec can reasonably capture the semantic meanings of tweets. We have modified the HMM in GMove by changing the observation state from a bag of keywords to a vector in low-dimensional space. Correspondingly, we selected VMF distribution to model the generation of a
tweet’s vector representation. We have derived the update rule for EM algorithm, which involves an approximation update rule to the extremely complicated VMF distribution. We have successfully achieved convergence for our new algorithm. We have conducted experiments to evaluated the performances of GMove and our model. The experiment was designed to predict a user’s next step movement given his or her previous steps. It turns out that our model has a similar accuracy with GMove. To analyze this result, we have conducted auxiliary experiments by removing all the text from tweets. To our surprise, the accuracy did not decrease a lot, which reveals the fact that text information is not critical in this specific task. We have also proposed other potential factors leading to the experimental results, such as the noisy and sparse characteristics of tweets. For future works, it will be interesting to try our model on other tasks (which have been discussed in Section 5) to see whether the model is more powerful or not.

Acknowledgements

This work is done under the supervision of Chao Zhang. We would like to give our special thanks to him for the great help. Also thanks to Prof. Jiawei Han, Xiang Ren, and Wenqi He for providing us comments and suggestions after the presentation, which helps a lot when writing the report.
References


[3] Zhenhui Li, Bolin Ding, Jiawei Han, Roland Kays, and Peter Nye. Mining periodic behaviors for moving objects. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1099–1108. ACM, 2010.


[12] Zhijun Yin, Liangliang Cao, Jiawei Han, Jiebo Luo, and Thomas Huang. Diversified trajectory pattern ranking in geo-tagged social media. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 980–991. SIAM, 2011.


[16] Chao Zhang, Keyang Zhang, Quan Yuan, Luming Zhang, Tim Hanratty, and Jiawei Han. Gmove: Group-level mobility modeling using geo-tagged social media. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1305–1314. ACM, 2016.


