T-Gram: A Time-Aware Language Model to Predict Human Mobility

Hsun-Ping Hsieh[1], Cheng-Te Li[2], Xiaoqing Gao[3]

[1]Graduate Institute of Network and Multimedia, National Taiwan University, Taipei, Taiwan
[2]Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
[3] School of Computer Science and Technology, Xidian University, China
d98944006@csie.ntu.edu.tw, ctli@citi.sinica.edu.tw, gaoxq@stu.xidian.edu.cn

Abstract

This paper presents a novel time-aware language model, *T-gram*, to predict the human mobility using location check-in data. While the conventional n-gram language model, which use the contextual co-occurrence to estimate the probability of a sequence of items, are often employed to predict human mobility, the time information of items is merely considered. T-gram exploits the time information associated at each location, and aims to estimate the probability of visiting satisfaction for a given sequence of locations. For a location sequence, if locations are visited at right times and the transitions between locations are proper as well, the T-gram probability gets higher. We also devise a *T-gram Search* algorithm to predict future locations. Experiments of human mobility prediction conducted on Gowalla check-in data significantly outperform a series of n-gram-based methods and encourage the future usage of T-gram.

Introduction

Location-based services (LBS), such as Foursquare and Gowalla, keep track of personal geospatial journeys through check-in actions. With smart phones, users can easily perform check-in actions, and the geographical information of locations with timestamps is stored in LBS. A large-scaled user-generated location sequences (i.e., routes) data are derived. Such location sequence data can not only collectively represent the real-world human geo-activities, but also serve as a handy resource for constructing location-based recommendation systems. Since the user-moving records implicitly reveal how people travel around an area with rich spatial and temporal information, including longitude, latitude, and recording timestamp, one can use such data to model human mobility by inferring the next locations of users.

Existing work on modeling human mobility has two directions. One is *location recommendation* (e.g. Ye et al. 2011), which is to recommend new locations that users have never visited before. The other is *location prediction* (e.g. Monreale et al. 2009), which is to predict the next existing locations that users had ever visited. The approach to these tasks is using n-gram language models with the consideration of the historical locations visited by users. However, the visiting time information hidden in locations and the time duration transit between locations are merely investigated.

We think the time information is important for predicting human mobility. First, the reasonability and pleasure of visiting a place are significantly affected by the visiting time. The user satisfaction would be diminished if going at inappropriate time. To either enjoy better experiences, people might be believed to move and stay at places with the right time. Second, the time spent on transitions between locations limit the human mobility. It is impossible for people to use a short time to transit between places distant to each other, and unreasonable to take lots of hours to transit between locations with a few city blocks to one another.

This paper proposes a novel time-aware language model, *T-gram*, to model and predict the human mobility using location check-in data. Comparing to conventional language models, which capture the contextual correlation between consecutively visited locations, T-gram uses the time information associated at each location, and aims to estimate the probability of visiting satisfaction for a given sequence of locations associated with visiting time stamps. The idea is that for a location sequence, if its locations are visited at their right times and the transitions between locations are proper as well, the T-gram probability gets higher. In other words, T-gram can be regarded as a measure that determines how well the locations along a location sequence are visited in terms of the visiting time and transition time. Based on *T-gram*, we develop a predictive method, *T-gram Search*, to predict the human mobility whose goal is to predict the future locations of a user.

Related Work

The relevant studies on predicting human mobility using historical location sequences can be divided into two categories: *location prediction* and *location recommendation*. Location prediction focuses on predicting the next existing

locations that the user had ever visited while location recommendation is to recommend new locations that the user has never visited before. The general consideration of location prediction and recommendation is the historically geolocations visited by users, and the common solution is using probabilistic language models or Markov models to capture the spatio-temporal correlation between locations so that the locations can be predicted successively.

Location Prediction

Monreale et al. (2009) predict the next location of a moving object with mined frequent trajectory patterns that capture the common paths. They construct a decision tree-like structure, *T-pattern Tree*, as a predictor of the next location of a new trajectory finding the best matching path in the tree. Ying et al. (2011) further leverage the *semantic* information, which describes the activities (in the form of *tags* and *types*) of locations. Then given the recent moves of a user, they define and compute the matching score geographically and semantically between mined frequent sub-trajectories and the given moves to find the best matched trajectory for the next location prediction. Sadilek et al. (2012) predict the most likely location of an individual, given the historical trajectories of his/her friends. They exploit the discrete *dynamic Bayesian network* to model the motion patterns of Twitter users from their friends, in which there is a hidden target user node and a number of observed information nodes of locations in each time slice. In addition to the query time, Chiang (2013) further consider the current time to predict locations. They construct a *Time-constrained Mobility Graph* that captures a user's moving behavior within a certain time interval, and compute the *reachability* between locations to infer next one.

Location Recommendation

Ye et al. (2011) point out the importance of *geographical influence*, which refers to that people tend to visit (a) *nearby* locations and (b) may be interested in *farther* locations that they are in favor of, to predict check-in locations. Liu and Liu et al. (2013) perform *personalized* point-of-interest recommendation. Their main advantage lies in using matrix factorization to mine the transition patterns of user preferences over location categories to enhance the prediction accuracy. Yuan et al. (2013) further propose a *time-aware* POI recommendation to recommend suitable locations for a certain user at a specified time in a day, using a temporal-enhance collaborating filtering. Noulas et al. (2012) predict the next check-in venue that a mobile user will visit by formulating it as a *ranking* task: given a user with his/her current check-in venue, ranking all the venues so that the predicted one is at the highest position. They devise the ranking measure considering transitions between types of places and flows of movement between places.

T-gram Model

We devise a novel time-aware language model, *T-gram*, to estimate the probability of a location sequence with time stamps. T-gram consists of two parts: (a) measuring the user satisfaction of visiting a location at a given time, and (b) measuring the reasonability when using a particular transition time between locations. We define the *location time distribution* of a location and *transition time distribution* between locations, which can be derived using the check-in time in the location sequences containing location l_i.

Definition 1. (Location Time Distribution). A *Visiting Time Distribution* (LTD) of location l_i is a probability distribution over time labels in hour, denoted by $LTD_{li}(t) = <(t_0, p_0), (t_1, p_1), ..., (t_{23}, p_{23})>$, where $p_0 + p_1 + ... + p_{23} = 1.0$.

Definition 2. (Transition Time Distribution). A *Transition Time Distribution* (TTD) between location l_i and l_j is defined as the probability distribution over time duration Δ in hour, $TTD_{li,lj}(\Delta) = <(\Delta_1, p_1), (\Delta_2, p_2), ..., (\Delta_{23}, p_{23})>$, where $p_1 + p_2 + ... + p_{23} = 1.0$.

T-gram Language Model

The proposed T-gram language model consists of two parts: measuring the goodness of the visiting time of locations and the transition time between locations using *LTD* and *TTD* respectively. We elaborate the details of *LTD* pleasant measure. Note that the pleasant measure of *TTD* follows the same settings. Assuming we want to know how well a decision is to visit a place at time t, given the location's *LTD*, we propose to first generate a thin Gaussian distribution $G(t;\mu,\sigma^2)$ whose mean value μ is at time t with a very small variance σ^2 (e.g. standard deviation is 1). And then we can transform the original task into measuring the difference between the Gaussian distribution with the learnt *LTD* of such location. We use the *symmetric Kullback-Leibler* (*KL*) *Divergence* between $G(t;\mu,\sigma^2)$ and $LTD_l(t)$ to represent the fitness of the assignment. The formal mathematical definition of a fitness score between a place *l* and a time *t* can be defined as:

$$D_{KL}(G(t;\mu,\sigma^2)||LTD_l(t))$$
$$= \sum_x G(x;\mu,\sigma^2)\, log\frac{G(x;\mu,\sigma^2)}{LTD_l(x)} + \sum_x LTD_l(x)\, log\frac{LTD_l(x)}{G(x;\mu,\sigma^2)}$$

Conceivably, a smaller *KL* value indicates better match between the assignment and the distribution learned from data. Consequently, we formally define the *T-gram* probability, $Prob_T(s)$, of a location sequence $s=<(l_1,t_1), (l_2,t_2), ..., (l_n,t_n)>$, as a combination of the popularity of places together with the fitness of each location over time, in the following equation.

$$Prob_T(s) =$$
$$\alpha \times \left(\prod_{i=1}^{n} D_{KL}\big(G(t;t_i,\sigma^2)||LTD_{l_i}(t)\big) \times \frac{1}{pop(l_i)}\right)^{\frac{-1}{n}} +$$

$$(1-\alpha)\times\left(\prod_{i=1}^{n-1} D_{KL}\left(G(t;\Delta_{i,i+1};\sigma^2)||TTD_{l_i l_j}(\Delta)\right)\right)^{\frac{-1}{n-1}}$$

where $pop(l_i) = N(l_i)/N_{max}$, $N(l_i)$ is the number of recording actions performed on location l_i, and N_{max} is the maximum number of recording actions among all the locations in the location sequence dataset. In addition, the parameter α is used to control the preference or importance on either visiting time or transition time. Higher α values refer to prefer the visiting quality on locations while lower α values indicate the human mobility should be strictly scheduled based on regular transportations. If the places in a location sequence *s* are visited during the proper time period, the $Prob_T(s)$ value would become higher.

Application to Predict Human Mobility

We aim to use the proposed T-gram language model to predict human mobility in check-in data. Given the source-destination query Q_d = (l_s, t_s, l_d) that depicts the clue of human mobility to be predicted, where l_s is the initial location, t_s is the starting timestamp, l_d is the destination location, and/or the preferred number of locations *k* if any, the goal of the mobility prediction is to predict the location sequence in response to Q_s or Q_d. We propose a *T-gram Search* algorithm to predict mobility. The basic idea is to find a path s = <(l_1=l_s,t_1=t_s), (l_2,t_2), ..., (l_k=l_d,t_k)> among locations such that the value of $Prob_T(s)$ is maximized.

Input: (a) the location sequence data *LS*; (b) the mobility query Q_d = (l_s, t_s, l_d); (c) *k*: the number of locations.
Output: the predicted $s = \langle(l_s, t_s), (l_2, t_2), \ldots (l_d, t_k)\rangle$

1: $s_0 = \langle(l_1 = l_s, t_1 = t_s)\rangle$.
2: $PriorityQueue = \{(s_0, 0)\}$.
3: $s = s_0$.
4: $l_{last} = null$.
5: **while** $|s| < k$ **do:**
6: $l_{last} = s_r.endLocation$.
7: $C = \{l_{next} | l_{last} \rightarrow l_{next}$ in $LS\}$.
8: **for each** $l_c \in C$ **do:**
9: **for each** $t_c \in T_{l_c}$ **do:**
10: $s_{tmp} = s + \langle(l_c, t_c)\rangle$.
11: $score = Prob_T(s_{tmp})$.
12: $PriorityQueue.Insert((s_{tmp}, score))$.
13: $s = PriorityQueue.Pull()$.
14: **return** *s*.

Figure 1: *T-gram Search* Algorithm.

T-gram Search consists of three steps, and the detailed algorithm is described in Figure 1. We first construct the initial location sequence s_0 by including the initial location l_s (line 1). A $PriorityQueue$ is used to maintain the location sequence with the highest T-gram score (line 2). Each element in $PriorityQueue$ consists of a location sequence *s* and the corresponding T-gram score. $PriorityQueue$ automatically sorts its elements according to their T-gram scores. We add s_0 to initialize $PriorityQueue$. After setting the final location sequence s_r as the initial one s_0 (line 3), we perform the iterative expansion search process until the location sequence *s* is constructed up to length *k* (line 5-13). Each iteration the last location l_{last} in the location sequence *s* with the highest T-gram probability is identified (line 6 and line 13) and each possible next visiting location l_{next} from the location sequence data is put into a candidate set *C* (line 7). Then for each candidate next location l_c, and for each time label t_c in the time label set T_{l_c} of l_c, we can derive the score $Prob_T(s_{tmp} = s + \langle(l_c, t_c)\rangle)$e. We put $Prob_T(s_{tmp})$ together with the corresponding location sequence s_{tmp} into $PriorityQueue$ (line 8-12). $PriorityQueue$ will then pick the next best location sequence and location to conduct the further expansions (line 13). Finally (line 14), the location sequence *s* is reported as the predicted one.

Experiments

We use a large-scale check-in data from Gowalla (Cho et al. 2011) for the experiments. This Gowalla dataset contains 6,442,890 check-in records from Feb. 2009 to Oct. 2010. The total number of check-in locations is 1,280,969. By constraining a location sequence as consecutive check-in locations of a user within a day, we can obtain 1,136,737 location sequences whose lengths are more than one and the average length of location sequence is 4.09. We extract two check-in subsets falling into the urban areas of New York and San Francisco. Some statistics are reported in Table 1.

Table 1: Statistics of two check-in data subsets.

	Total Number of Check-ins	Avg. Seq Length	Number of Locations
New York	103,174	4.46	21,973
San Francisco	187,568	4.09	15,406

Experiment 1: **Location Sequence Ranking.** We aim to verify if the proposed *T-gram* model can rank the location sequences higher than the fake ones. We first randomly choose one thousand real location sequences. For each location sequence, we replace a portion of locations with other locations in the same city to generate a pseudo sequence. Each selected location sequence is paired with a pseudo one. To make the task non-trivial, we adopt a replacing strategy to replace a location with a plausible one instead of a randomly selected one. That is, to replace a location at position *i* of a location sequence, we only choose from candidate locations that have ever appear right after the location at position *i*-1 (e.g. the bigram probability of them is non-zero). We use T-gram model to examine each pair of the real location sequence and its pseudo sequence, and record how frequently our method ranks the correct one higher. We report the accuracy of our method and compare it with baselines. The accuracy is defined as the number of correctly ranked pairs (i.e. an algorithm assigns

higher score to the real location sequence than the pseudo sequence) divided by the total number of pairs.

Baseline Methods. We also design the following baseline competitors, which search in the check-in database in a greedy manner to find the location sequence that not only satisfies query requirement, but also maximize a certain objective function $f(r)$. (a) *Distance-based Approach* chooses the closest location to the current spot as the next spot to move to. It measures the quality a location sequence by using the objective tion $f_{dist}(r) = \sqrt[n]{\prod_{i=1}^{n}(1/D(l_i, l_{i-1}))}$, where $D(l_i, l_{i-1})$ is the geographical distance between locations. (b) *Popularity-based Approach* chooses the most popular spot of a given time to be the next spot. It rates the location sequence using the goodness function$f_{pop}(r) = \sqrt[n]{\prod_{i=1}^{n}(N(l_i)/N_{max})}$. (c) *Forward Heuristic Approach* chooses location l_i possessing the largest bi-gram probability with the previous location $P(l_i|l_{i-1})$ as the next: $f_{forw}(r) = \sqrt[n]{P(l_1)P(l_2|l_1)P(l_3|l_2)\cdots P(l_n|l_{n-1})}$. (d) *Backward Heuristic Approach* chooses l_i that possesses the largest bi-gram probability with the next location $P(l_i|l_{i+1})$ as the next location. The goodness function is designed as$f_{backw}(r) = \sqrt[n]{P(l_1|l_2)P(l_2|l_3)\cdots P(l_{n-1}|l_n)}$.

We vary the number of replaced locations from 10% to 50%. We set the parameter α in T-gram as 0.5. Figure 2(a) shows the results for San Francisco. T-gram model achieves around 98% accuracy in distinguishing the real location sequences from fake ones. The forward and backward heuristics reaches about 80% accuracy. The popular-based and distance-based methods do not do a good job as they only reach 50% or lower in accuracy. Similar trend happens in New York (Figure 2(b)). The results are not surprising because T-gram considers the location preference over time.

Experiment 2: Mobility Cloze Test. Given a set of location sequences with time stamp in each location, through randomly removing *m* consecutive locations in each location sequence, we aim to test whether a method can successfully recover the missing locations. With the increasing of *m*, consecutive removal of locations does impose decent level of difficulty to this cloze test. It is because that when *m* increases, the information that can be utilized becomes sparse, and mistakes in the earlier position can lead to follow-up errors in the next positions to be predicted. We use *Hit Rate* as the accuracy measure for the cloze test. Given there are totally *N* removals of locations over all location sequences, and assumed *M* places out of *N* is successfully predicted, the hit rate is defined as *M*/*N*. Higher hit rate indicates better quality of recommendation. Note that when there are multiple missing places in the cloze test, we only consider the fully-recovered sequences as hits. By varying the number of missing instances per location sequence and report the hit rates. The results are shown in Figure 3. In

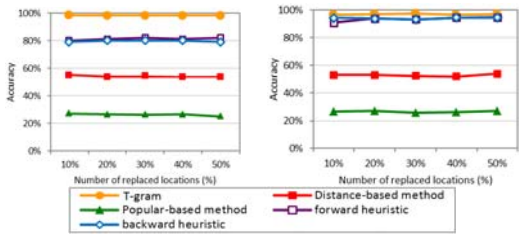

Figure 2: Accuracy by varying the number of replaced locations in San Francisco (left) and New York (right).

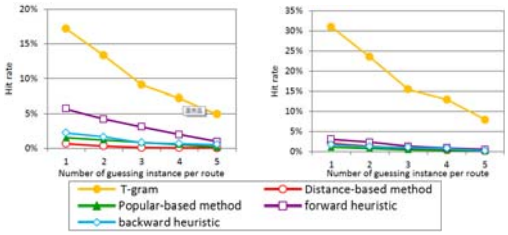

Figure 3: Hit rate by varying the number of guessing instance per sequence in San Francisco (left) and New York (right).

general, the hit rates are decreasing while the number of missing instance increases. Such results encourage the usage of T-gram with the predictive method to model human mobility.

Conclusion

In this paper, we propose a novel time-aware language model, T-gram, to measure the probability of visiting satisfaction for a given time-stamped location sequence. We also develop a T-gram Search algorithm to predict the human mobility. Experiment results show that T-gram outperforms a series of n-gram-based methods on location sequence ranking and mobility cloze test.

References

M.-F. Chiang, Y.-H. Lin, W.-C. Peng, and P. S. Yu. Inferring distant-time location in low-sampling-rate trajectories. In *ACM KDD* 2013.

E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: User movement in location-based social networks. In *ACM KDD* 2011.

B. Liu, Y. Fu, Z. Yao, and H. Xiong. Learning Geographical Preferences for Point-of-Interest Recommendation. In *ACM KDD* 2013.

X. Liu, Y. Liu, K. Aberer, and C. Miao. Personalized point-of-interest recommendation by mining users' preference transition. In *ACM CIKM* 2013.

A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. Where next: a location predictor on trajectory pattern mining. In *ACM KDD* 2009.

A. Noulas, S. Scellato, N. Lathia, and C. Masolo. Mining User Mobility Features for Next Place Prediction in Location-based Services. In *IEEE ICDM* 2012.

A. Sadilek, H. Kautz, and J. P. Bigham. Finding your friends and following them to where you are. In *ACM WSDM* 2012.

M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *ACM SIGIR* 2011.

J.-C. Ying, W.-C. Lee, T.-C. Weng, and V. S. Tseng. Semantic trajectory mining for location prediction. In *ACM GIS* 2011.

Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Time-aware point-of-interest recommendation. In *ACM SIGIR* 2013.