Efficient Fine-grained Location Prediction Based on User Mobility Pattern in LBSNs

Jiuxin Cao, Shuai Xu, Xuelin Zhu, Renjun Lv, Bo Liu
School of Computer Science and Engineering
Southeast University
Jiulonghu, Nanjing, China
Email: { jx.cao, xushuai7, zhuxuelin, lvrenjun, bliu } @seu.edu.cn

***Abstract*—Location-Based Social Networks (LBSNs) have built bridges between virtual space and real-world mobility in recent years. The massive check-in data generated in LBSNs has made it possible to predict users' future check-in location, which has proved meaningful for e-commerce developments. Existing studies mainly focus on predicting the next check-in location with a coarse granularity, which shows limited performance in practical scenarios. In this paper, we propose a comprehensive approach based on user mobility pattern for predicting users' future check-in location at any fine-grained time in LBSNs. Firstly, user mobility patterns involving time periodicity, global popularity and user preference are analyzed. Then, a set of predictive features are extracted. Finally, the features are combined into a supervised scoring model and a classification model respectively in order to predict users' future check-in location. Extensive experiments on three real-world datasets demonstrate the efficiency and superiority of the proposed approach in various metrics.**

***Keywords*-LBSN; Location Prediction; Mobility Pattern; Scoring Model; Classification Model**

I. Introduction

The inflating prevalence of mobile Internet hastens the popularity of **L**ocation-**B**ased **S**ocial **N**etworks (**LBSN**s), among which some typical platforms like Foursquare have become indispensable applications on the Internet. Integrating users' online behavior with offline mobility through location, LBSNs make it possible to explore intrinsic pattern of user mobility. The massive check-in data generated by millions of users in LBSNs can be used to predict their future check-in location based on their mobility pattern and historical check-in records. This application scenario has proved valuable for traffic planning, product recommendation and disaster warning, to name a few. In the scenario of product recommendation, for example, an efficient location prediction approach can significantly improve user experience in location-based services and help them avoid the cumbersome process of selecting products from mass options.

Research on location prediction has become a hot issue over these years. Earlier studies [1–4] are mainly about predicting the next check-in location for users. These techniques tend to show limited performance in practical scenarios as they cannot deal with situations in the far future, while the latter is more valuable in business applications. For a few studies [5–8] that deal with predicting check-in location long in advance, they often pay attention to mining individual factors separately to construct prediction algorithms and ignore the integration effect of various factors on user mobility. Besides, existing studies mainly carry out location prediction with a coarse granularity concerning location category [2, 8, 9] or single day as the time measurement [10]. Apparently, the practicability of such location prediction approaches is narrowed in the increasingly fierce competition environment for location-based services.

In view of the shortcomings resided in current studies, we aim to predict users' future check-in location at any fine-grained time accurate to hour based on a comprehensive consideration of time periodicity, global popularity and user preference. In essence, our goal is to predict users' future check-in probability at a location where he has visited before. To fulfill this task, we have to construct a heterogeneous social network model according to the fundamentals in LBSNs, and give a formal definition of the location prediction problem.

The main contributions of this paper are summarized as follows:

- We analyze user mobility pattern and extract a set of predictive features involving time periodicity, global popularity and user preference.
- We train a supervised scoring model to evaluate the possibility of a given user's visit to a candidate location at a given time, thus we can integrate the features to improve prediction performance.
- We reduce the location prediction problem to a binary classification task and train a tree-based model to classify whether a given user would check-in at a candidate location at a given time.
- We verify the performance of the proposed approach through extensive experiments on three real-world LBSN datasets, and demonstrate its superiority over six baseline methods.

The remainder of the paper is organized as follows. Section II introduces the heterogeneous social network model and formalizes the location prediction problem. In Section III, a set of predictive features are extracted. In Section IV, individual features are combined for improving prediction performance. The classification task for location prediction

978-1-5386-1072-5/17 $31.00 © 2017 IEEE
DOI 10.1109/CBD.2017.48

is described in Section V, followed by experimental evaluation in Section VI. Finally, Section VII concludes the paper.

II. Model Construction and Problem Definition

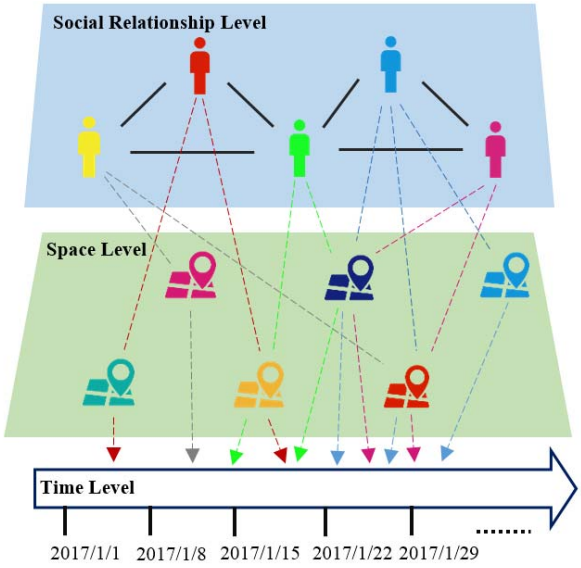

Figure 1. Framework of the LBSN model

The framework of the constructed LBSN model, which consists of three levels, i.e. social level, space level and time level, is depicted in Fig.1. Conventionally, a LBSN model can be represented in a quadruple shaped as $G < U, V, E, C >$, where U stands for node set meaning users, V stands for location set, E represents edge set and C stands for the historical check-in set. If a user u has made n check-ins in his historical records, then the check-in set of u can be expressed as $C_u = \{< u, v_1, t_1 >, < u, v_2, t_2 >, ..., < u, v_n, t_n >\}$. Based on the LBSN model and its representation, we further define user $u's$ friend set as $F_u = \{u' | (u, u') \in E_{UU}\}$ and $u's$ location set as $V_u = \{v | (u, v) \in E_{UV}\}$.

For each user u_i, assume that he has visited L_i locations, then we can formalize the fine-grained location prediction problem as follows. Given a time t that accurate to hour, we rank all the L_i locations so that the exact location u_i will visit at time t is ranked at the highest possible position in the ranking list. The definition of location prediction problem in this paper can also be stated as: given a time t and a specific location v, judging whether u_i will check-in at v at time t. The two versions of definition correspond to ranking problem and classification problem, respectively. We will put forward two strategies to solve these problems in this paper.

III. Feature Extraction

A. Time Periodicity

Adequate studies like [11] show that human mobility has a strong time periodicity. Day mode and Week mode are proposed in [12] to depict the time periodicity of user mobility pattern. We follow this method in [12] and map the absolute time into discrete time windows under both Day mode and Week mode. For Day mode, $r_d(t) = \{0, 1, 2, ..., 23\}$. For Week mode, $r_w(t) = \{0, 1, 2, ..., 167\}$.

For a given user u and time t_{given} under Week mode, the feature $Week\_loc\_1$ of candidate location $v_{candidate}$ can be quantified as formula (1):

$$Week\_loc\_1 = \frac{|\{c|(c.v = v_{candidate}) \cap (r_w(c.t) = r_w(t_{given})), c \in C_u\}|}{|\{c|r_w(c.t) = r_w(t_{given}), c \in C_u\}|} \quad (1)$$

Indeed, $Week\_loc\_1$ is equivalent to the historical check-in frequency at $v_{candidate}$ under Week mode. Similarly, we can define the feature Day_loc_1 of $v_{candidate}$ under Day mode.

As users do not always check-in at a fixed time, we enlarge the given time t_{given} to a 3-hour interval. Under this situation, for a given user u and time t_{given} under Week mode, the feature $Week\_loc\_3$ of candidate location $v_{candidate}$ can be quantified as formula (2):

$$Week\_loc\_3 = \frac{|\{c|(c.v = v_{candidate}) \cap (|r_w(c.t) - r_w(t_{given})| \leq 1), c \in C_u\}|}{|\{c\,|r_w(c.t) = r_w(t_{given}), c \in C_u\}|} \quad (2)$$

The feature Day_loc_3 of candidate location $v_{candidate}$ can also be quantified in the same way.

Locations in LBSNs are often tagged with semantic information like category such as 'Food' and 'Shop & Service'. User check-in distribution over different categories under different time modes usually varies with each other. Based on this observation, we define the feature $Week\_cate$ of candidate location $v_{candidate}$ under Week mode as formula (3):

$$Week\_cate = \frac{|\{c|(z.v = z.v_{candidate}) \cap (r_w(c.t) = r_w(t_{given}), c \in C_u\}|}{|\{c\,|r_w(c.t) = r_w(t_{given}), c \in C_u\}|} \quad (3)$$

where $z.v$ denotes the category to which location v belongs.

Similarly, we can quantify the check-in frequency on $z.v_{candidate}$ under Day mode as Day_cate.

B. Global Popularity

As [13] indicates, users' check-in locations hold a clustering property in space, namely, there exists a check-in center for each user, and the closer a user is to that center, the higher probability he will check-in there. In this paper, we propose to discover user check-in center as follows. Firstly, we cluster a user's check-in locations using DBSCAN algorithm, and select a cluster with maximum locations; then, we average the latitude and longitude of check-ins in that cluster and get a mean coordinate; in the end, we calculate the average value of latitude and longitude of all the user's check-in locations within a radius of 10km

to that point. In this way, we take the final coordinate as the user's check-in center.

Based on the observation that a user often visits places close to his check-in center, we define the global feature $Distance$ as formula (4):

$$\textit{Distance} = dist(v_{candidate}, center) \tag{4}$$

where $dist(v_i, v_j)$ means the geographic distance from location v_i to location v_j, while $center$ indicates the check-in center of this user.

Meanwhile, we extract two other global features $Loc\_pop.$ and $Cate\_pop.$ performed by all users in U, which measure the global popularity of $v_{candidate}$ and $z.v_{candidate}$, respectively.

C. User Preference

We confirm that each user has his own preference for certain location and certain category, therefore we extract two preference features $User\_loc\_pref.$ and $User\_cate\_pref.$ to capture the user's historical check-in frequency at $v_{candidate}$ and $z.v_{candidate}$, respectively.

The last feature to be extracted is $Friendship$. We treat the influence of friends on user mobility as the social relationship feature, through which user similarity and friends' check-in records are simultaneously considered. The feature of $Friendship$ is computed in formula (5):

$$\textit{Friendship} = \sum_{u' \in F_u} [weight_{u'} * P(C_{u'} = v_{candidate})] \tag{5}$$

where $weight_{u'}$ measures the influence of friend u' on the user, and $P(C_{u'} = v_{candidate})$ is the check-in frequency of friend u' at candidate location $v_{candidate}$. In this paper, $weight_{u'}$ is calculated based on the similarity between u and u' using their common check-in locations.

IV. Supervised Scoring Model

A. Scoring Model

Due to the fact that different features perform diversely in location prediction task, it is reasonable to assign different weights to different features. The final weighted scoring model for $v_{candidate}$ at time t_{given} is defined as formula (6):

$$\zeta(u, t_{given}, v_{\text{candidate}}) = \Theta \cdot \Psi = \sum_{i=1}^{m} \theta_m * \varphi_m \tag{6}$$

where $\Psi = \{\varphi_1, \varphi_2, ...\varphi_m\}$ is the feature vector containing all the m feature values and $\Theta = \{\theta_1, \theta_2, ...\theta_m\}$ is the weight vector in which each element is the corresponding weight for each feature.

Based on the scoring model, we can compute the overall score of each candidate location $v_{candidate}$ at a given time t_{given}. Afterwards, we are able to rank all candidate locations based on their scores in descending order when predicting the check-in location for a given user u.

B. Parameter Inference

In order to learn parameter Θ for feature vector Ψ , we have to explain the partition of user check-in records for model training in the first place. For each user, his check-in records are split into two parts according to the size ratio 9:1 in time order. The former 90% records are used to compute individual feature values, while the latter 10% records are used for testing.

For a user u_i and each check-in record in his former 90% check-in data, we retrieve a positive sample s_i^1 by computing m feature values for the check-in location at the check-in time, and assign this sample with a positive label. Then a negative sample s_i^0 can be obtained by randomly selecting a location where u_i does not visit at that time. Note that the selected location for negative sample must be in the candidate location list of u_i. Based on the partition of check-in records, we can construct the training sample set R, in which each training sample is a tuple $< s_i^1, s_i^0 >$.

We further infer the parameter Θ through Maximum Likelihood Estimation (MLE) to ensure that for any tuple $< s_i^1, s_i^0 >$, the score of s_i^1 is larger than the score of s_i^0. Thus, we derive the optimization objective for the scoring model as follows in formula (7):

$$\begin{aligned} OF &= \arg\max_{\Theta} \ln p(R|\Theta)p(\Theta) \\ &= \sum_{u \in U} \sum_{<s_i^1, s_i^0> \in R_u} \ln \sigma(\varsigma(s_i^1) - \varsigma(s_i^0)) - \lambda ||\Theta||_2{}^2 \end{aligned} \tag{7}$$

where λ is a parameter for regularization.

Stochastic gradient descent algorithm is adopted in this paper to maximize the optimization objective OF. The gradient of OF to Θ can be computed in formula (8):

$$\frac{\partial OF}{\partial \Theta} \propto \sum_{u \in U} \sum_{<s_i^1, s_i^0> \in R_u} \frac{e^{-(\varsigma(s_i^1) - \varsigma(s_i^0))}}{1 + e^{-(\varsigma(s_i^1) - \varsigma(s_i^0))}} \frac{\partial}{\partial \Theta} (\varsigma(s_i^1) - \varsigma(s_i^0)) - \lambda \Theta \tag{8}$$

Then we can update Θ after each iteration according to formula (9) :

$$\Theta = \Theta + \alpha \left(\frac{e^{-(\varsigma(s_i^1) - \varsigma(s_i^0))}}{1 + e^{-(\varsigma(s_i^1) - \varsigma(s_i^0))}} \frac{\partial}{\partial \Theta} (\varsigma(s_i^1) - \varsigma(s_i^0)) - \lambda \Theta \right) \tag{9}$$

where α is the learning rate parameter. Note that when the maximum iteration number is fixed, we can match a best α for a specific λ through experiment.

V. Supervised Classification Model

To verify the comprehensive efficiency of the combined features on all users as a whole, we further reduce the ranking problem into a binary classification task.

First of all, we explain the construction of Training dataset and Testing dataset. All user check-in records are divided into three parts $D1$, $D2$, and $D3$ according to the size ratio 8:1:1 in time order. For each check-in record of user u_i

in $D1$, if he has a corresponding check-in location in $D2$, we retrieve a positive sample for that check-in record and assign this sample with a positive label $+1$; next, we retrieve all other check-in records that u_i has not visited in $D2$ as negative samples and assign these samples with negative label -1. In a similar way, we can construct the Testing dataset using all positive and negative samples generated by all users in $D1$+$D2$.

After the construction of Training and Testing datasets, a supervised binary classifier is trained on the Training dataset and subsequently evaluated on the Testing dataset. It is notable that the number of positive samples and negative samples is imbalanced in the Training dataset. In order to increase the sensitivity of the classifier to the minority samples, we adopt a method of over-sampling the minority samples which has been proposed in [14].

We consider two different classification models to fulfill this task, namely, tree-based classifier and logistic classifier. Experimental results prove that tree-based classifier (Random Forest) outperforms Logistic Regression classifier, so we will only report the experimental results using Random Forest in the next section.

VI. Experiments

A. Datasets

In this paper, we use three real-world Foursquare datasets for experimental evaluation. The first dataset **NYC** is crawled on our own while the other two datasets **NYC_pub** and **TKY_pub** are collected by [15]. Note that friendship information is originally missing in **NYC_pub** and **TKY_pub**. The statistics of the selected datasets are shown in Table I.

Table I
Statistics of Datasets

	NYC	***NYC_pub***	***TKY_pub***
# User	9,767	1,083	2,293
# Location	148,435	38,336	61,858
Avg. # Check-in	196	210	250
Avg. # Friend	8.2	*None*	*None*

B. Evaluation Metrics

For the scoring model, as it essentially corresponds to the ranking problem, we adopt $PercentileRank(PR)$ and $Accuracy@N$ as the evaluation metrics. The metric PR is defined as follows:

$$PR = \frac{|L| - rank(k)}{|L|} \tag{10}$$

where $|L|$ is the length of the candidate list, and $rank(k)$ is the position that a candidate location is ranked by our scoring model. The $Average\ PercentileRank(APR)$, obtained by averaging across all check-in predictions, measures the average normalized position of the correct predictions.

As for $Accuracy@N$ metric, we assess the predictive power of our approach using different prediction list size N. We successfully predict the future check-in location at a given time for a user only when we rank that location in the top-N list. Similar to APR, the mean $Accuracy@N$ is adopted.

In terms of the classification model, the evaluation metrics are usually $Precision$, $Recall$ and F1-$score$. Among the three metrics, F1-$score$ is the most comprehensive one since it synthetically evaluates the classifier by considering $Precision$ and $Recall$. Consequently, we mainly compare the F1-$score$ metric among all the methods to evaluate their classification performance.

C. Baseline Methods

- Most Frequent Check-in (MFC): we take MFC as a baseline method which is defined as $Global\ popularity + User\ preference$.
- Most Day Frequent (MDF): this baseline method only takes Day mode features into consideration.
- Most Week Frequent (MWF): this baseline method only takes Week mode features into consideration.
- Multi-feature Fusion (MfF) proposed in [3]: we mildly modify the next location prediction method to any-time location prediction method by converting geographical features in [3] into $Distance$ feature proposed in this paper.
- Community-based Location Prediction (CLP) [5]: this method predicts a user's future check-in location by selecting the most popular location in his community. Note that this method can not be applied on **NYC_pub** and **TKY_pub** due to the lack of friendship information.
- Geo-Recency Model (GRM) [16]: this model infuses geo-recency information into a Non-negative Matrix Factorization model, and uses the reconstructed user matrix for location prediction.

D. Results and Evaluation

To evaluate the prediction performance of the supervised scoring model, we must set the proper learning rate α, regularization coefficient λ and the number of iterations. We traverse α from 0.01 to 0.1 (+0.01 each time) and λ from 0.1 to 1 (+0.1 each time) according to [17]. Besides, the maximum iteration number is set to be 100,000. We report the best parameter combination for each dataset in Table II.

Table II
Experimental Parameter Setup

	Parameter Setup
NYC	$\alpha = 0.1$,$\lambda = 1$
NYC_pub	$\alpha = 0.04$,$\lambda = 0.1$
TKY_pub	$\alpha = 0.05$,$\lambda = 1$

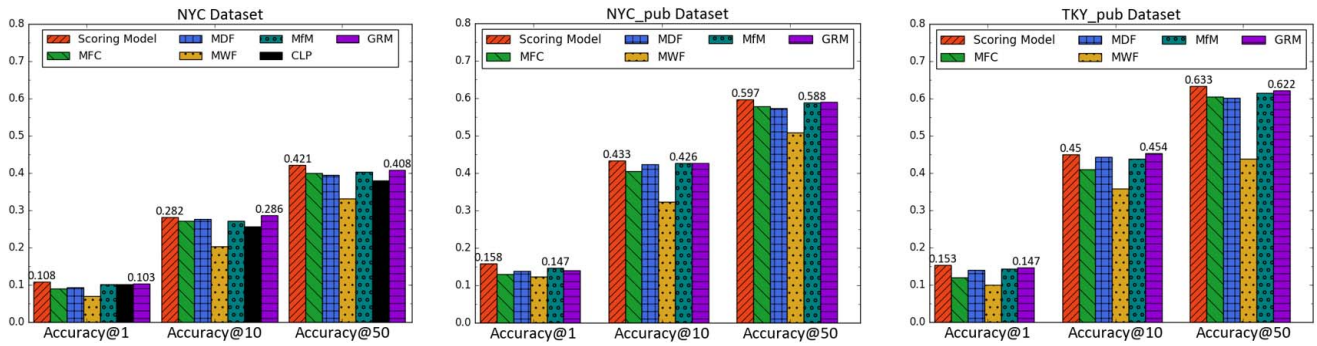

Figure 2. Mean Accuracy@N comparison results

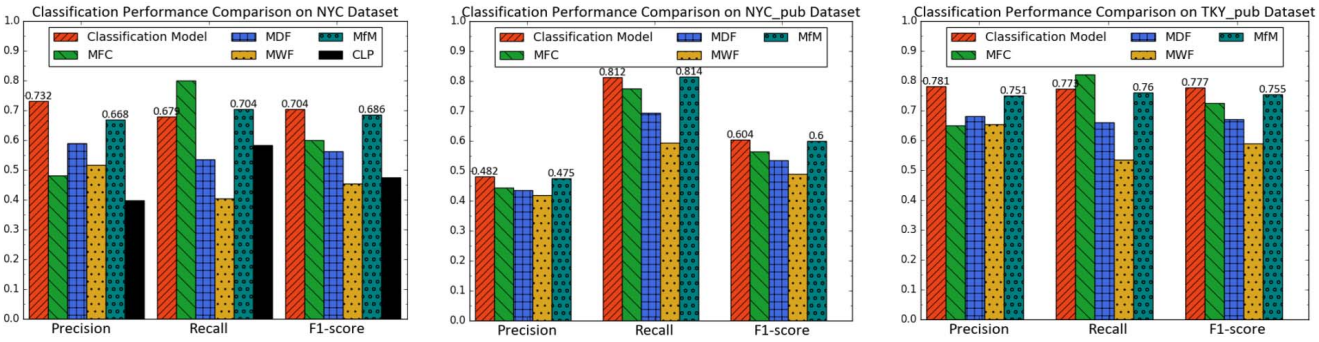

Figure 3. Classification performance comparison results

Firstly, we report the APR comparison results among various methods in Table III. As we can see, the scoring model achieves the best performance among all datasets. Compared with other methods, the scoring model outstands the best comparison methods with 1.8%, 1.3% and 1.3% on **NYC**, **NYC_pub** and **TKY_pub**, respectively. This observation indicates that our scoring model can rank the true location at a higher position in the candidate list than other baseline methods, which demonstrates its superiority in terms of the ranking problem.

Table III
APR COMPARISON RESULTS

	NYC	***NYC_pub***	***TKY_pub***
Scoring Model	**0.837**	**0.864**	**0.866**
MFC	0.815	0.834	0.836
MDF	0.804	0.833	0.835
MWF	0.686	0.718	0.741
MfF	0.822	0.853	0.855
CLP	0.794	–	–
GRM	0.807	0.847	0.852

Secondly, we report the mean $Accuracy@N$ results in Fig.2. As we can see, in most cases the supervised scoring model outperforms other baseline methods even though GRM proposed in [16] may exceed the scoring model on $Accuracy@10$ in **NYC** and **TKY_pub** datasets. It is notable that the scoring model dominates with both $Accuracy@1$ and $Accuracy@50$. Considering the practical application scenario that we can rank a location a user will visit at any time in a Top-1 and Top-50 position, it is a remarkable performance as there are tens of thousands of places to be ranked in a city.

To summarize, we assert that the supervised scoring model is able to obtain peak performance in both APR and $Accuracy@N$ metrics, which demonstrates the efficiency of the combined features.

Next, we compare the prediction performance of the classification model with baseline methods. The results are displayed in Fig.3. Note that the baseline method GRM can not be applied for classification because no predictive features can be extracted using this method. As is shown in Fig.3, the proposed classification model is superior to the baseline methods in terms of F1-$score$ though the advantage over MfF is not as obvious as over other methods. For F1-$score$, the performance of classification model is 2.6%, 1.2% and 2.9% higher than the best baseline method MfF on **NYC**, **NYC_pub** and **TKY_pub**, respectively.

In total, the performance of random forest classifier using the combined features largely exceeds the performance of baseline methods, showing that the feature combination approach proposed in this paper is more efficient when

addressed in a non-linear way.

VII. Conclusion and Future Works

In this paper, we study the problem of location prediction in Location Based Social Networks. Based on the constructed LBSN model, we firstly extract twelve individual features involving time periodicity, global popularity as well as user preference. Then, all the individual features are combined into a supervised scoring model to evaluate the possibility of a given user's visit to a candidate location. Thirdly, a classification model based on random forest is trained to classify whether a given user would check-in at a candidate location. Experimental results based on three real world datasets demonstrate the efficiency and superiority of our approach over baseline models.

As for future work, a dynamic user preference for location may be a supplement to the further study. Besides, information fusion based on multiple heterogeneous networks may help improve the performance of location prediction.

Acknowledgment

This work is supported by National Natural Science Foundation of China (61272531, 61202449, 61272054, 61370207, 61370208, 61300024, 61320106007 and 61472081), China high technology 863 program (2013AA013503), Jiangsu Technology Planning Program (SBY2014021039-10), Jiangsu Provincial Key Laboratory of Network and Information Security under Grant No.BM2003201 and Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grant No.93k-9.

References

[1] D. Lian, Y. Zhu, X. Xie, and E. Chen, "Analyzing location predictability on location-based social networks," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2014, pp. 102–113.

[2] A. Likhyani, D. Padmanabhan, S. Bedathur, and S. Mehta, "Inferring and exploiting categories for next location prediction," in *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015, pp. 65–66.

[3] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo, "Mining user mobility features for next place prediction in location-based services," in *Data mining (ICDM), 2012 IEEE 12th international conference on*. IEEE, 2012, pp. 1038–1043.

[4] W. Li, C. Eickhoff, and A. P. de Vries, "Want a coffee?: predicting users' trails," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2012, pp. 1171–1172.

[5] J. Pang and Y. Zhang, "Exploring communities for effective location prediction," in *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015, pp. 87–88.

[6] R. Assam and T. Seidl, "Check-in location prediction using wavelets and conditional random fields," in *Data Mining (ICDM), 2014 IEEE International Conference on*. IEEE, 2014, pp. 713–718.

[7] Y.-S. Cho, G. Ver Steeg, and A. Galstyan, "Where and why users" check in"." in *AAAI*, 2014, pp. 269–275.

[8] E. Bart, R. Zhang, and M. Hussain, "Where would you go this weekend? time-dependent prediction of user activity using social network data." in *ICWSM*, 2013.

[9] J. Ye, Z. Zhu, and H. Cheng, "What's your next move: User activity prediction in location-based social networks," in *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 2013, pp. 171–179.

[10] R. Zhang, B. Price, M. Chu, and A. Walendowski, "Location-based predictions for personalized contextual services using social network data," in *First Workshop on recent advances in behavior prediction and pro-active pervasive computing*. Citeseer, 2012.

[11] S. M. Rahimi and X. Wang, "Location recommendation based on periodicity of human activities and location categories," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2013, pp. 377–389.

[12] H. Gao, J. Tang, X. Hu, and H. Liu, "Exploring temporal effects for location recommendation on location-based social networks," in *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 2013, pp. 93–100.

[13] C. Petersen, J. G. Simonsen, and C. Lioma, "Power law distributions in information retrieval," *ACM Transactions on Information Systems (TOIS)*, vol. 34, no. 2, p. 8, 2016.

[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[15] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 129–142, 2015.

[16] R. Assam, S. Sathyanarayana, and T. Seidl, "Infusing geo-recency mixture models for effective location prediction in lbsn," in *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 2016, pp. 855–863.

[17] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 2009, pp. 452–461.