Large-scale Point-of-Interest Category Prediction Using Natural Language Processing Models

Daniel (Yue) Zhang*, Dong Wang*, Hao Zheng*, Xin Mu†, Qi Li*, Yang Zhang*
**Department of Computer Science and Engineering*
† Department of Aerospace and Mechanical Engineering
University of Notre Dame
Notre Dame, IN, USA
yzhang40@nd.edu, dwang5@nd.edu, hzheng3@nd.edu, xmu@nd.edu, qli8@nd.edu, yzhang42@nd.edu

***Abstract*—Point-of-Interest (POI) recommendation is an important application in Location-based Social Networks (LBSN). The *category prediction* problem is to predict the next POI category that users may visit. The predicted category information is critical in large-scale POI recommendation because it can significantly reduce the prediction space and improve the recommendation accuracy. While efforts have been made to address the POI category prediction problem, several important challenges still exist. First, existing solutions did not fully explore the temporal dependency (e.g., "long range dependency") of users' check-in traces. Second, the hidden contextual information associated with each check-in point has been underutilized. In this work, we propose a Context-Aware POI Category Prediction (CAP-CP) scheme using Natural Language Processing (NLP) models. In particular, to address temporal dependency challenge, we develop a novel Temporal Adaptive Ngram (TA-Ngram) model to capture the dynamic dependency between check-in points. To address the challenge of hidden context incorporation, CAP-CP leverages the Probabilistic Latent Semantic Analysis (PLSA) model to infer the semantic implications of the context variables in the prediction model. Empirical results on a real world dataset show that our scheme can effectively improve the performance of the state-of-the-art POI recommendation solutions.**

***Keywords*-POI Recommendation, Context-Aware Prediction, Natural Language Processing, Location-based Social Networks**

I. Introduction

This paper presents a new Point-of-Interest (POI) category prediction scheme to improve the accuracy of large-scale POI recommendation using natural language processing (NLP) models. POI recommendation is an important application in LBSNs where the goal is to provide personalized POI recommendations to users. Significant efforts have been made to address the POI recommendation problem in data mining, information retrieval and recommendation systems [1], [2], [3], [4], [5], [6], [7].

However, there exists a fundamental problem that has not been well addressed by the current POI recommendation solutions: the prediction space of the POI is huge (e.g., hundreds of thousands of POIs in large cities) but the data from each user on the LBSNs is often sparse (e.g., due to the lack of incentives or privacy concerns). This leads to a poor performance of the personalized POI recommendations [8]. The user's preference on the POI category has been recently shown to be effective in reducing the prediction space and improving the recommendation performance [6]. However, the user's preference may evolve over time and her/his future preference is unknown *a priori* in real world LBSNs [9], [10]. Therefore, this paper focuses on a *POI category prediction* problem where the goal is to predict the next POI category that users may visit.

While efforts have been made to address the POI category prediction problem [3], [6], several important technical challenges still exist. The first challenge is "temporal dependency". While it has been proven that historical check-in traces can be used to predict user's future check-ins [3], existing solutions using Markov models [1], [11], [12] suffer from several fundamental limitations. First, the Markov property assumes that the current check-in point is only dependent on the recent check-ins and fails to capture "long-range dependency". For example, a user's next visit may depend on last year's visit on the same day (e.g., anniversary celebration) rather than his/her visits in the past few days. Second, the dependency between consecutive check-ins from the Markov models may not be strong in real world scenarios especially when the time gap between two consecutive check-ins is large. For example, an infrequent LBSN user can upload his first check-in at a restaurant and his next check-in at a movie theater one month later. These two consecutive check-ins might have a very weak or even no dependency at all.

The second challenge is hidden context incorporation. Several methods have been proposed to incorporate context information into the solution of the POI recommendation problems [13], [2], [14]. However, these methods only model the *direct correlation* between context (e.g., time, location, weather) and the POI category while ignoring the underlying user's decision process (i.e., hidden context). For example, these solutions fail to answer the question on "*why* would Alice make a decision to go to a bar rather than stay at home on Saturday evenings ?". Another related problem with context incorporation is the "curse of dimensionality" where the incorporation of hidden contexts into the prediction problem will increase the dimension of search space, which can potentially cause the over-fitting problem when the context space is large [15].

In this paper, we develop a Context-Aware POI Category Prediction (CAP-CP) scheme that uses two core NLP language models (i.e., Ngram and PLSA) to address the above challenges. In particular, to address the *temporal dependency* challenge, we develop a Temporal Adaptive Ngram (TA-Ngram) to dynamically decide the length and strength of dependency between check-in points. To address the *hidden context incorporation* challenge, we apply the Probabilistic Latent Semantic Analysis (PLSA) to jointly model the latent semantic setting, user decision pattern and the context to predict the user's next POI category. We evaluate our scheme on a real world dataset collected from Foursquare. The evaluation results show that our scheme significantly improve the POI recommendation performance by accurately predicting the POI categories that users may visit in the near future.

II. Related Work

Similar works have been proposed to leverage POI category information to improve the POI recommendation results. For example, Ye *et al.* developed a mixed Hidden Markov Model to estimate the next category of user's activity and predict the most likely POI given the estimated category distribution [3]. Liu *et al.* proposed a category-aware POI recommendation model that exploits the transition patterns of user's preference over location categories to improve the POI recommendation accuracy [6]. However, a common limitation of these methods is that they all assume Markov property in category transition and thus suffer from "long-range dependency" problem. In contrast, the CAP-CP scheme incorporates a novel temporal adaptive Ngram model to address the "long-range dependency" problem.

Previous works have also considered the context information in the POI recommendation [16], [5], [4]. For example, Gao *et al.* investigated the temporal patterns of check-ins in terms of temporal non-uniformness and temporal consecutiveness in POI recommendations [16]. Yuan *et al.* incorporated both temporal cyclic information and geographical information for time-aware POI recommendations [5]. However, these approaches focus on the explicit relationships between the context information and POIs while ignoring the underlying decision process of users (i.e., hidden context). In our work, we explicitly model the user decision process with latent semantic analysis to understand "why" the user makes a certain decision in a specific context.

III. Problem Formulation

In this section, we formulate our context-aware POI category prediction problem. Consider a LBSN application where a set of I users $U = \{U_1, U_2, ..U_I\}$ voluntarily report their check-in points at venues[1] in a city. A check-in point consists of the location, the category of the visited venue and the check-in timestamp. We define a set of X possible categories of POIs $CP = \{cat_1, cat_2, ..., cat_X\}$, where cat_x denotes the x^{th} category of POI. For the i^{th} user, we define his/her historical check-in trace as $Trace(i) = \{p_i^1, p_i^2, ..., p_i^{K(i)}\}$ where p_i^k is the POI category of the k^{th} check-in point from U_i and $K(i)$ is the total number of check-ins that user U_i has provided.

We further define a set of M context variables related to the check-in traces. Examples of such context variables include *spatial context* (e.g., the user's current location), *temporal context* (e.g., day of the week, time of the day), *natural context* (e.g., weather conditions) and *social context* (e.g., social events and festivals). Without losing of generality, we use $C = \{C1, C_2, ..., C_M\}$ to define the set of all possible context variables we consider in our model. For user U_i, we define $Context(i) = \{f_i^1, f_i^2, ..., f_i^{F(i)}\}$ where f_i^k is a vector of contexts associated with the k^{th} check-in point from U_i and $F(i)$ denotes all distinct values in $Context(i)$.

Take Foursquare as an example. Each user account is considered as a user. Examples of POI category from check-in traces include "restaurant", "outdoor", "transportation", "art & entertainment" and "shops & services". If we consider "spatial", "temporal", "natural" and "social" dimensions of the context, an example of C is {"user's location", "day of the week", "weather condition", "social events"} and an example f_i^k can be {"home", "weekend", "cloudy", "football game"}.

The objective of the user POI category prediction is to predict a user's next category as accurate as possible:

$$\arg\max_x Pr(cat_x = p_i^{K(i)+1} | Context(i), Trace(i)) \quad (1)$$

where cat_x is the predicted POI category that U_i would visit next.

IV. Solution

A. Overview of CAP-CP Scheme

Our CAP-CP system consists of two key components: TA-Ngram model and PLSA model. First, we develop the TA-Ngram model to predict a user's next POI category based on the "syntactic pattern" of the user's historical check-in traces using the Ngram model. Second, we propose the PLSA predictor to incorporate the contextual information and the semantic implication of the context variables. Finally we aggregate the results of TA-Ngram and PLSA models to accomplish the prediction task.

B. Temporal Adaptive Ngram Predictor

In our POI category prediction problem, we map "words" to POI categories and "Ngram" to a sequence of N consecutive POI categories (which is referred to as "sequences" in

[1] We use the term venue and POI interchangeably in the rest of the paper.

the rest of this paper). We build a “training corpus” which is a list of sequences and their counts (i.e., the number of occurrences) in the check-in traces. We then use the “training corpus” to predict the next POI category using the Ngram model. In particular, the n-th check-in category can be predicted as:

$$Pr(w_n|w_{n-1}^1) \approxeq Pr(w_n|w_{n-1}^{n-N+1}) \tag{2}$$

where w_i denotes the i^{th} POI category in the trace and w_j^i denotes a sequence of consecutive POI categories $\{w_i, w_{i+1}...w_j\}$ (i.e., w_j^i is an Ngram when $j = i+N-1$).

We observe a few technical challenges of applying the basic Ngram model to solve the POI category prediction problem: i) the user check-in trace is usually sparse and the time gap between two consecutive check-ins is sometimes long (e.g. several months). The dependency assumption between consecutive check-ins made by Ngram might not always hold; ii) the basic Ngram model ignores the “freshness” of check-ins: the recent check-in categories may be more relevant to the next POI category compared to the old ones.

To make Ngram model more suitable to our POI category prediction problem, we develop a Temporal-Aware Ngram model (TA-Ngram). In particular, to remove the non-existing dependency between consecutive check-ins, we insert a breaking character (like “period” in a sentence) between category w^{i-1} and w^i when the time gap between them is too large. We exclude any category sequence that contains breaking character from the training corpus. To capture the “freshness” of the POI category sequence, we use the following function to adjust the weights on the counts of a sequence in the training corpus. For an occurrence of POI sequence $seq(t)$ at the time interval t, we compute its weight as:

$$weight(seq(t)) = e^{\frac{t}{\alpha}}, t \in Inv \tag{3}$$

where Inv denotes the total number of time intervals of the check-in trace and the α is a tuning parameter. We then apply Witten-Bell smoothing algorithm to handle unseen sequences [17].

C. Latent Semantic Analysis to Explore Context Awareness

An inherent limitation of the Ngram based model is the “long-range dependency” where the model fails to explore the dependency between check-ins that are far apart from each other (i.e., beyond of the size of N). To address this problem, we developed a Probabilistic Latent Semantic Analysis (PLSA) based model. In particular, the PLSA explores the semantic meaning of the training corpus to compensate for the “long-range dependency” limitation of Ngram [18]. The PLSA model fits nicely into our problem because i) it can capture the latent factors that affect users' preferences on POI visit within different contexts; ii) it reduces the high dimensional space of the context variables to a lower dimensional latent semantic space, which helps alleviate the curse of dimensionality.

We first define a set of latent semantic settings $Z = \{z_1, z_2, ..., z_{|Z|}\}$ to represent a group of hidden factors that lead to the user's intention to visit a POI category under a specific context. For example, a context of “Friday night” may represent a semantic setting of “party”, which contributes to the user's decision to go to nightlife places. Similarly, a “Monday morning” may represent the setting of “work”, which may contribute to the user's decision to visit public transportation. The idea of PLSA model is illustrated in Figure 1.

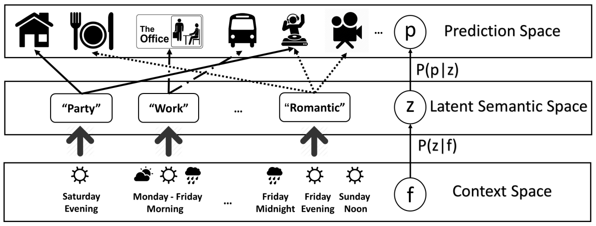


Figure 1: PLSA Model for POI Category Prediction

To simplify our notations, we denote a POI category as p, a context vector as f, and a semantic setting as z. The likelihood of observing the co-occurrences of p and f in the check-in trace can be written as:

$$\begin{aligned} L(p, f) &= \prod_{p \in P} \prod_{f \in F} Pr(p, f)^{n(p,f)} \\ &= \prod_{p \in P} \prod_{f \in F} (\sum_{z \in Z} Pr(f) Pr(p|z) Pr(z|f))^{n(p,f)} \end{aligned} \tag{4}$$

where $n(p, f)$ is the count of the co-occurrences of p and f in the trace. $Pr(p|z)$ denotes the probability that a user decides to visit category p given a latent semantic setting z and $Pr(z|f)$ denotes the probability of the setting z given a context f. P and F denote the sets of all distinct POI categories and context vectors for a user. The estimation parameters $\theta_{PLSA} = \{Pr(p|z), Pr(z|f)\}$ can be derived using the EM algorithm [19]. Finally, we predict a user's next POI category as:

$$Pr(p|f) = \sum_{z \in Z} Pr(p|z) Pr(z|f) \tag{5}$$

D. User Regularity Detection

The TA-Ngram model works better for the check-in traces with strong regularity (i.e. a user periodically visits a set of POI categories) and the PLSA model works better for the check-in traces with more randomness (less regularity). To “get the best of both worlds”, we design a User Regularity Detection (URD) module to detect the regularity of users' check-in traces.

We develop a URD module to jointly detect the regularity of a check-in trace and identify the optimal size N for the

TA-Ngram scheme. In particular, we first define a sequence mapping function $Map(ws, seq)$ to convert the check-in trace to a sequence of Ngrams, where $ws \in [1, N]$ is the window size for scanning and seq is the user check-in trace. The output of the function is a sequence of Ngrams with size ws by sliding the window through the check-in trace. In the previous example, with $ws = 2$, the check-in sequence "Food, Movie, Food, Movie, Food, Movie" is mapped into a new sequence "111" (1 denotes the bigram "Food, Movie"). After the conversion, the entropy is calculated for the Ngram sequence.

To find the optimal size of the TA-Ngram, we calculate the window size for user i's check-in trace ws_i that gives the minimum entropy of the sequence (i.e., the sequence that is most regular) as follows.

$$\begin{aligned} N = ws_i &= \arg\min_n H(Map(n, Trace(i))) \\ H(seq) &= -\sum_{pos_j \in seq} Pr(pos_j) log(Pr(pos_j)) \end{aligned} \tag{6}$$

where pos_j denotes the Ngram at the j^{th} position. We denote the normalized minimum entropy for user i as NH_i. We use this value to decide how "regular" the check-in trace is.

E. Final Scheme

We integrate all components presented earlier into a holistic CAP-CP system. Let $PLSA(i, K)$ and $TAN(i, N)$ denote the output of the PLSA and TA-Ngram models for user U_i. K is the number of topics in PLSA and N is the size of TA-Ngram. Our final prediction on $Pr(p|f)$ for user U_i is calculated as:

$$Pr(p|f) = \begin{cases} TAN(i, ws_i), NH_i \leq thres_l \\ (1-\beta) * TAN(i, ws_i) + \beta \\ \quad * PLSA(i, K), thres_l < NH_i < thres_h \\ PLSA(i, K), NH_i \geq thres_h \end{cases} \tag{7}$$

The intuition is that we use the TA-Ngram predictor when a user's check-in trace is identified as regular (NH_i is small enough) and use PLSA predictor when the trace is identified as irregular (NH_i is big enough). When the trace is a mixture of regular and irregular sequences, we combine the results of the two predictors.

V. Evaluation

A. Data Collection and Pre-Processing

To evaluate the performance of the CAP-CP scheme, we use a real-world Foursquare data trace [2] collected from New York City (NYC) between Apr. 3rd 2012 and Sep. 16 2013 [20]. The datatrace has a total of 1,083 users, 38,333 venues and a total of 227,428 check-ins. We select three context variables for our PLSA predictor in the evaluation. We define and process these context variables as follows:

- **Weather:** We collect historical daily weather information from Weather Wunderground [3] that covers the data collection period of each data trace.
- **Day of the Week:** We label each check-in as happening on either "weekday" or "weekend" based on the date of the check-in.
- **Time of the Day:** We label each check-in as happening in "Morning" (6:00-10:59), "Noon" (11:00 -12:59), "Afternoon" (13:00 - 16:59), "Evening" (17:00 - 21:59), "Night" (22:00 - 5:59) based on the timestamp of the check-in.

Finally, we use the nine categories (i.e. Arts & Entertainment, College & University, Food, Outdoors, Nightlife, Professional & Other Places, Residence, Shop & Service, Travel & Transport) defined by the official Foursquare developer documentation [4] as the candidate classes for our prediction task.

B. Baseline Methods

We choose the following representative POI category prediction schemes as our baselines.

- **ST-LDA:** A latent probabilistic generative model that learns region-dependent personal interest for POI recommendations [9].
- **CIKM13:** A category-aware POI recommendation model that exploits the transition patterns of users' preference over location categories in POI recommendations [6].
- **Order k Markov:** It models the category transition process using simple Markov Chains with order k varying from one to three [3].
- **TA-Ngram:** Our proposed TA-Ngram predictor with N varying from two (bi-gram) to six.
- **PLSA:** Our proposed PLSA predictor with the topic number varying from one to ten.

We sort the check-ins in a chronological order for each user and select the first 80% of check-ins as the training data and the remaining 20% as the test data for the prediction evaluation. The parameters are tuned using the data from the training set. We set the time gap as 1 week for breaking character, the time interval as 1 month to capture freshness, and $|Z| = 5$, $thres_l = 0.2$, $thres_h = 0.8$ for all datasets. For a fair comparison, we chose the parameter assignment that yields the best result for baselines that have a tunable parameter.

C. Evaluation on Real World Data Traces

1) Evaluation Metrics: In the evaluation, we use the following evaluation metrics: prediction accuracy, Precision@K, and Recall@K. The prediction accuracy is given by $Accuracy = \frac{\sum_{j \in L} TP_j + TN_j}{\sum_{j \in L} TP_j + TN_j + FN_j + FP_j}$ where TP_j, TN_j,

[2] https://sites.google.com/site/yangdingqi/home/foursquare-dataset

[3] www.wunderground.com/history/

[4] developer.foursquare.com/categorytree

FP_j and FN_j represents True Positives, True Negatives, False Positives and False Negatives respectively for the target category j, and L denotes the set of all categories. Precision@K and Recall@K are the standard evaluation metrics in POI recommendations [5]. In particular, Precision@K defines the ratio of successfully predicted labels (i.e., POI categories) to the K recommendations, and Recall@K defines the ratio of successfully predicted labels to the total number of labels to be predicted.

	Accuracy	Precision	Recall
CAP-CP	**0.4915**	**0.2599**	**0.7796**
ST-LDA	0.4372	0.2324	0.6973
CIKM13	0.4173	0.2383	0.7150
Markov	0.2456	0.1785	0.5355
TA-Ngram	0.3771	0.2342	0.7026
PLSA	0.3267	0.1744	0.5232

Table I: Category Prediction Performance of All Schemes

2) Evaluation Results: Table I shows the performance results of all compared schemes. We observe that our CAP-CP scheme significantly outperforms other baselines on all evaluation metrics. This is due to the fact that the baseline schemes either fail to capture the temporal dependencies between check-ins or fail to incorporate the hidden contextual information in their prediction model. The recall and precision are evaluated based on top 3 recommendations which are chosen considering the prediction space of 9 categories. We also evaluate the performance by varying the size of recommendation list and CAP-CP continues to outperform other baselines (Figure 2). The non-trivial performance gains of CAP-CP on both precision and recall demonstrate that our scheme can predict the next POI category with fewer guesses than other baselines.

Confusion Matrix: CAP-CP

Label	**Predicted**								
	Art	Col.	Food	Nig.	Out.	Res.	Pro.	Shop	Tra.
Art	**0.38**	0.04	0.21	0.03	0.03	0.07	0.03	0.07	0.15
Col.	0.07	**0.45**	0.11	0.05	0.03	0.02	0.03	0.09	0.15
Food	0.04	0.02	**0.47**	0.03	0.03	0.04	0.02	0.10	0.25
Nig.	0.04	0.04	0.04	**0.57**	0.02	0.04	0.07	0.10	0.07
Out.	0.09	0.04	0.24	0.05	**0.32**	0.07	0.03	0.09	0.09
Res.	0.07	0.00	0.08	0.05	0.02	**0.52**	0.05	0.09	0.12
Pro.	0.06	0.07	0.06	0.04	0.06	0.09	**0.49**	0.03	0.10
Shop	0.05	0.03	0.08	0.03	0.03	0.03	0.03	**0.59**	0.14
Tra.	0.03	0.02	0.06	0.03	0.01	0.02	0.03	0.09	**0.72**

Confusion Matrix: ST-LDA

Label	**Predicted**								
	Art	Col.	Food	Nig.	Out.	Res.	Pro.	Shop	Tra.
Art	**0.33**	0.03	0.35	0.03	0.04	0.08	0.04	0.03	0.07
Col.	0.05	**0.42**	0.17	0.00	0.00	0.02	0.14	0.12	0.08
Food	0.06	0.01	**0.52**	0.02	0.05	0.09	0.06	0.12	0.08
Nig.	0.05	0.08	0.14	**0.22**	0.03	0.03	0.19	0.14	0.14
Out.	0.06	0.00	0.30	0.01	**0.29**	0.12	0.06	0.11	0.05
Res.	0.10	0.01	0.06	0.04	0.03	**0.55**	0.05	0.08	0.09
Pro.	0.03	0.02	0.21	0.03	0.04	0.08	**0.45**	0.06	0.10
Shop	0.06	0.02	0.39	0.03	0.07	0.07	0.02	**0.28**	0.07
Tra.	0.03	0.04	0.05	0.03	0.03	0.00	0.02	0.03	**0.77**

Table II: Confusion Matrix for CAP-CP and ST-LDA

To more closely investigate the performance gain achieved by CAP-CP scheme, we consider not only whether the prediction is correct but also which categories are confused with each other. Table II shows confusion matrices for CAP-CP and ST-LDA, one of the best performing baselines from the previous experiment. We observe that CAP-CP outperforms the ST-LDA in correctly predicting the majority of the correct category of POIs and minimizing possible confusions (i.e., large numbers of diagonal elements and small numbers in others). For example, we observe that ST-LDA has trouble recognizing the Shop category (28%) and confuses it with Food. This is because that user's traces on these two categories often mingle together, making it hard for the ST-LDA to make the right prediction. In contrast, CAP-CP has a high accuracy for the Shop category (59%) thanks to the hidden context incorporation using PLSA in our scheme (e.g., users are more likely to go to shopping venues on "weekends" or "sunny/cloudy days"). The context information provides extra evidence to avoid confusion between similar categories.

VI. Conclusion

This paper develops a new POI category prediction scheme (i.e. CAP-CP) using NLP models. In particular, we develop a TA-Ngram model that captures the dynamic dependency between check-in points from user's trace to address the temporal dependency challenge. We develop a PLSA based model that explores the latent semantic settings in the user's visit decision process to address the hidden context incorporation challenge. Empirical results on a real-world data trace show that the CAP-CP scheme can accurately predicting the user's category preference in the future. The results of this paper are important because they offer a new set of approaches to better solve the large-scale POI recommendation problem in particular and context-aware prediction problems in general using NLP models.

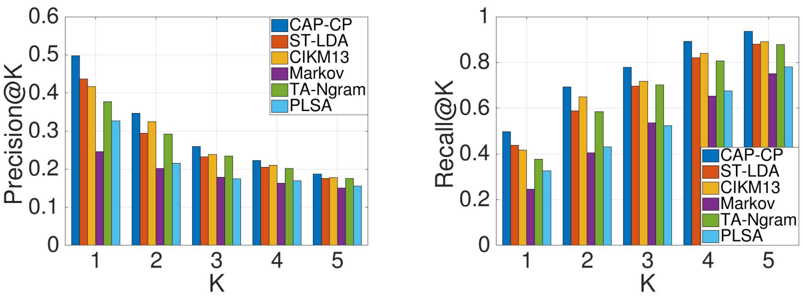
Acknowledgment

This research is supported in part by the National Science Foundation under Grant No. CBET-1637251, CNS-1566465 and IIS-1447795 and Army Research Office under Grant W911NF-17-1-0409. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

[1] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation." in *IJCAI*, vol. 13, 2013, pp. 2605–2611.

[2] Z. Yao, Y. Fu, B. Liu, Y. Liu, and H. Xiong, "Poi recommendation: A temporal matching between poi popularity and user regularity," in *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 2016, pp. 549–558.



(a) NYC Precision@K

(b) NYC Recall@K

Figure 2: POI Recommendation Precision@K and Recall@K of All Compared Schemes

[3] 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.

[4] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011, pp. 325–334.

[5] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, "Time-aware point-of-interest recommendation," in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2013, pp. 363–372.

[6] X. Liu, Y. Liu, K. Aberer, and C. Miao, "Personalized point-of-interest recommendation by mining users' preference transition," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 2013, pp. 733–738.

[7] C. Huang and D. Wang, "On interesting place finding in social sensing: An emerging smart city application paradigm," in *Smart City/SocialCom/SustainCom (SmartCity), 2015 IEEE International Conference on*. IEEE, 2015, pp. 13–20.

[8] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, "Personalized ranking metric embedding for next new poi recommendation." in *IJCAI*, 2015, pp. 2069–2075.

[9] H. Yin, X. Zhou, B. Cui, H. Wang, K. Zheng, and Q. V. H. Nguyen, "Adapting to user interest drift for poi recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 10, pp. 2566–2581, 2016.

[10] D. Zhang, H. Rungang, and D. Wang, "On robust truth discovery in sparse social media sensing," in *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE, 2016.

[11] D. Y. Zhang, C. Zheng, D. Wang, D. Thain, X. Mu, G. Madey, and C. Huang, "Towards scalable and dynamic social sensing using a distributed computing framework," in *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*. IEEE, 2017, pp. 966–976.

[12] D. Zhang, D. Wang, and Y. Zhang, "Constraint-aware dynamic truth discovery in big data social media sensing," in *2017 IEEE International Conference on Big Data (IEEE BigData 2017)*. IEEE, 2017.

[13] Y. Liu, C. Liu, B. Liu, M. Qu, and H. Xiong, "Unified point-of-interest recommendation with temporal interval assessment." in *KDD*, 2016, pp. 1015–1024.

[14] D. Wang, T. Abdelzaher, and L. Kaplan, *Social sensing: building reliable systems on unreliable data*. Morgan Kaufmann, 2015.

[15] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE transactions on pattern analysis and machine intelligence*, vol. 19, no. 2, pp. 153–158, 1997.

[16] 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.

[17] I. H. Witten, A. Moffat, and T. C. Bell, *Managing gigabytes: compressing and indexing documents and images*. Morgan Kaufmann, 1999.

[18] X. Wang, A. McCallum, and X. Wei, "Topical n-grams: Phrase and topic discovery, with an application to information retrieval," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 697–702.

[19] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999, pp. 50–57.

[20] D. Yang, D. Zhang, and B. Qu, "Participatory cultural mapping based on collective behavior in location based social networks," *ACM Transactions on Intelligent Systems and Technology*, vol. 7, no. 3, pp. 467–479, 2015, in press.