Sparse User Check-in Venue Prediction By Exploring Latent Decision Contexts From Location-Based Social Networks

Daniel (Yue) Zhang, Yang Zhang, Qi Li, and Dong Wang

Abstract—The proliferation of online Location-Based Social Networks (LBSN) has offered unprecedented opportunities for understanding fine-grained spatio-temporal behaviors of users and developing new location-aware applications. In this work, we focus on the problem of "Sparse User Check-in Venue Prediction" where the goal is to predict the next venue LBSN users will visit by exploiting their sparse online check-in traces and the latent decision contexts. While efforts have been made to predict users' check-in traces on a LBSN, several important challenges still exist. First, check-in traces contributed by LBSN users are often too sparse to provide sufficient evidence for a reliable prediction, especially when the prediction space is huge (e.g., hundreds of thousands of venues in large cities). Second, the user's decision context on which venue to visit next is often latent and has not been incorporated by current venue prediction models. Third, the dynamic and non-deterministic dependency between check-ins is either ignored or replaced by a simplified "consecutiveness" assumption in existing solutions, leading to sub-optimal prediction results. In this work, we develop a *Context-aware Sparse Check-in Venue Prediction (CSCVP)* scheme inspired by natural language processing techniques to address the above challenges. In particular, CSCVP predicts the venue category information and explores the similarity between users to address data sparsity challenge by significantly reducing the prediction space. It also leverages the Probabilistic Latent Semantic Analysis (PLSA) model to incorporate the user decision context into the prediction model. Finally, we develop a novel Temporal Adaptive Ngram (TA-Ngram) model in CSCVP to capture the dynamic and non-deterministic dependency between check-ins. We evaluate CSCVP using three real-world LBSN datasets. The results show that our scheme significantly improves accuracy (30.9% improvement) of the state-of-the-art user check-in venue prediction solutions.

Keywords—User Check-in Venue Prediction, LBSN, Natural Language Processing, Data Sparsity, Decision Context

1 Introduction

This work is motivated by the emergence of Location-Based Social Network (LBSN) applications where users can voluntarily share their check-in traces (i.e., a sequence of visited venues) through online social platforms. Examples of LBSNs include Foursquare, Google Places, Yelp, and WeChat [1]–[3]. User check-in venue prediction is an important problem where the goal is to provide an accurate prediction of a user's next check-in venue on LBSN. The solution to this problem can contribute to various LBSN based applications such as targeted advertisement, Point-of-Interest (POI) recommendation, and user mobility profiling. Moreover, the accuracy to this problem is also critical to better user experience and engagement, for instance, streamlining the user experience for finding a venue, and for sending relevant push notifications for users who are highly sensitive to spam. Significant efforts have been made to address similar problems in data mining, information retrieval and recommendation systems [4]–[12]. Examples of such solutions include Collaborative Filtering (CF) [8], [13]–[15], Matrix Factorization (MF) [10], [12], and Markov Chain models [5], [7], [16]. However, several important limitations exist in current solutions. In particular, this paper addresses three specific challenges in the user check-in venue prediction problem: *data sparsity*, *latent decision context incorporation*, and *dynamic and non-deterministic temporal dependency*.

Data sparsity: a fundamental challenge that has not been well addressed by the current check-in venue prediction solutions is data sparsity - the prediction space is huge (e.g., hundreds of thousands of venues in large cities) but the data from each user on the LBSNs is often sparse. Several factors contribute to this challenge: i) the information sharing on LBSN is voluntary and users may lack incentives to continuously contribute data to the application [17], [18]; ii) fresh users may have very few check-ins when they just start to use the application; iii) users may refrain from posting all his/her check-ins online due to the privacy concerns [19], [20]. In fact, only around 16 check-ins per user are available in one of our datasets collected from Foursquare over a period of 17 months. Such sparse data can significantly degrade the performance of current check-in prediction approaches [7], [21], [22]. The challenge becomes more significant for the solutions that build a personalized prediction model for individual users (i.e., locally trained model).

Latent decision context incorporation: the second chal-

• *The authors are with the Department of Computer Science and Engineering and the Interdisciplinary Center for Network Science and Applications (iCeNSA), University of Notre Dame, Notre Dame, IN 46556. E-mail: {yzhang40,yzhang42,qli8,dwang5}@nd.edu*

lenge is the context in which a user makes the decision on which venue to visit next is often not directly observable from the check-in traces on LBSN. Several methods have been proposed to incorporate contextual information into the solutions of the check-in venue prediction problems [11], [23]–[25]. However, these methods only model the *direct correlation* between observable contexts (e.g., time, location, weather) and the check-in venue while ignoring the underlying user's decision process (i.e., *latent decision 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 latent 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 [26], [27].

Dynamic and non-deterministic temporal dependency: the third challenge is the dependency between consecutive check-ins of a user can be highly dynamic and non-deterministic. In previous studies, temporal dependency has been widely used for the prediction of a user's future check-ins [7]. However, existing solutions model temporal dependency mainly based on the "consecutiveness" assumption: the current check-in of a user depends on the previously consecutive check-ins [5], [28] or on the check-ins within a certain period of time [29]. However, such an assumption cannot be simply applied in a sparse prediction scenario due to the following reasons: i) consecutive check-ins may not necessarily imply the temporal dependency. For example, an infrequent LBSN user can upload his/her first check-in at a restaurant and his next check-in at a movie theater a few months later. These two consecutive check-ins might not have any temporal dependency at all; ii) the "consecutiveness" assumption largely ignores the fact of "long-range dependency" where check-ins that are far apart on the temporal dimension may still have strong dependencies (e.g., a user may always go to the same restaurant for an anniversary celebration) [30].

In this paper, we develop a Context-aware Sparse Check-in Venue Prediction (CSCVP) scheme inspired by the NLP language models to address the above challenges. We summarize our approach below.

- To address the *data sparsity* challenge, we decompose the check-in venue prediction problem into two sub-problems: (1) predicting the category of a user's next check-in venue (Section 4.2 - Section 4.5); and (2) predicting the most likely check-in venue given the estimated category distributions (Section 4.6). The key advantage of this approach is that it significantly reduces the prediction space. A user similarity module is also developed to further improve the prediction results by leveraging peer's influence.
- To address the *latent decision context incorporation* challenge, we apply the Probabilistic Latent Semantic Analysis (PLSA) to jointly model the latent semantic setting, user decision pattern and the rich contextual information (e.g., weather, geolocation, time of the day) to predict the user's future check-in venues (Section 4.3).
- To address the *dynamic and non-deterministic temporal dependency* challenge, we develop a Temporal Adaptive Ngram (TA-Ngram) to dynamically decide the length and strength of dependency between check-ins that do not follow the "consecutiveness" assumption (Section 4.2).

We evaluate our scheme on three real-world datasets collected from Foursquare. The extensive evaluation results show that our scheme significantly improves the user check-in venue prediction performance compared to the state-of-the-art baselines.

A preliminary version of this work has been published in [31] to investigate the *venue category prediction* problem. The current paper is a significant extension of the previous work in the following aspects. First, we solved a new problem of sparse user check-in venue prediction problem in this paper where the goal is to predict the actual *venue* instead of the category of the venue the user will visit next. This new problem is more challenging than the venue category prediction problem because the prediction space for the venue is much larger than the category [7]. Second, we developed a new solution CSCVP that explicitly addresses data sparsity challenge by exploring the similarities between users and peer influence (Section 4). Third, we develop a new principled approach based on a co-training framework to integrate results from the TA-Ngram and PLSA predictors to predict the next venue category a user will visit (Section 4). Fourth, we perform a set of new experiments on two newly added real-world datasets from Foursquare (i.e., Tokyo and Paris) to evaluate the performance and robustness of the CSCVP scheme in different scenarios (Section 5). Fifth, we compared our scheme with three new baselines from recent literature and show the consistent performance improvements achieved by the CSCVP scheme (Section 5). Finally, we extended the related work by adding the discussion on more recent works about user check-in venue prediction and topic modeling techniques in NLP (Section 2).

2 Related Work

User check-in venue prediction in LBSN provides the foundation for various applications such as POI recommendation [32], user mobility analysis [7] and targeted advertising [10]. Previous studies have made significant progress to address this problem [5], [10]–[12], [23]. The most commonly used techniques for user check-in venue prediction are Collaborative Filtering (CF) and Matrix Factorization (MF). For example, Ye *et al.* proposed a user based CF algorithm that considers the social and spatial influence using a Bayesian model [8].

Zheng *et al.* proposed three collective tensor and matrix factorization models to provide personalized venue and activity recommendations [33]. Liu *et al.* combined matrix factorization and users' preference transition to provide personalized next venue prediction [10]. Li *et al.* proposed a Spatial-Temporal Probabilistic Matrix Factorization Model (STPMF) that models a user's preference for venue as the combination of his/her geographical preference and other general interests [12]. However, the above approaches suffer from the problem of huge check-in venue prediction space and sparse data on LBSN as we discussed in the introduction [9], [28]. In contrast, our CSCVP system addresses this problem by inferring the category distribution of the user's next check-in venue, which greatly reduces the prediction space and improves the prediction accuracy.

Previous works have also considered the contextual information in predicting users' future check-in preferences [8], [9], [11]. For example, Liu *et al.* developed a unified framework to model the joint effect of multiple factors such as user preferences, geographical influences, and user mobility behaviors [23]. Gao *et al.* investigated the temporal patterns of check-ins in terms of temporal non-uniformness and temporal consecutiveness in venue recommendations [29]. Yuan *et al.* incorporated both temporal cyclic information and geographical information for time-aware check-in venue prediction [9]. However, these approaches focus on the explicit relationships between the contextual information and check-ins while ignoring the underlying decision process of users (i.e., latent decision context). In our work, we explicitly model the user decision context with latent semantic analysis to understand "why" the user makes a certain check-in decision in a specific context.

Several recent works have also started to leverage the check-in category information to improve the check-in venue prediction performance. 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 check-in given the estimated category distribution [7]. Liu *et al.* proposed a category-aware check-in prediction model that exploits the transition patterns of user's preference over location categories to improve the check-in venue prediction accuracy [10]. Sang *et al.* developed a venue category transition centric prediction scheme where successive check-ins can be predicted based on the category transition probabilities [34]. However, a common limitation of these methods is that they assume "consecutiveness" in category transition and thus suffer from "long-range dependency" problem. Also, they do not explicitly consider the data sparsity problem in their solutions. In contrast, the CSCVP scheme incorporates topic modeling (i.e., PLSA) to capture the "long-range dependency" of check-ins and combines it with a novel user similarity regulation model to further alleviate data sparsity challenge.

There exist some similarities between our work and previous studies on LBSN venue prediction using topic modeling techniques. For example, Yin *et al.* proposed a probabilistic generative topic model to learn region dependent user preference that can adapt to user interest drift across geographical regions and improve user mobility prediction [23]. Jiang *et al.* developed an author topic model-based collaborative filtering method for personalized travel recommendations by clustering users based on similar topic preferences (e.g., cultural, cityscape, landmark) [22]. Kurashima *et al.* used PLSA to mine probabilistic photographer behavior for travel route recommendations [16]. However, the above approaches do not consider the syntactic structure of time-series data (e.g., the temporal dependency between check-ins) and thus can hardly be applied to our problem of user check-in venue prediction. In contrast, our scheme provides a principled approach that considers both syntactic (using TA-Ngram) and semantic (using PLSA) features of check-in data to effectively predict the user's preference on the check-in venues.

3 Problem Formulation

In this section, we formulate our sparse user check-in venue prediction problem on LBSN. We assume a sparse data scenario where users may only contribute a very limited amount of check-ins as compared to the actual venues they have visited. Consequently, the time gap between consecutive check-ins can be large. Formally, we consider a LBSN application where a set of I users $U = \{U_1, U_2, ..U_I\}$ voluntarily report their check-in points at set of venues. For the i^{th} user, we define his/her historical check-in venue trace as $\mathcal{V}(i) = \{v_i^1, v_i^2, ..., v_i^{K(i)}\}$ where v_i^k is the venue 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. A check-in point consists of the GPS location of the check-in venue, the category of the venue, and the check-in timestamp. Similarly, we define a check-in category trace as $\mathcal{P}(i) = \{p_i^1, p_i^2, ..., p_i^{K(i)}\}$ where p_i^k is the category of v_i^k.

We further define a set of M context variables related to the check-in venues. 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 loss of generality, we use $C = \{C_1, C_2, ..., C_M\}$ to define the set of all possible context variables we consider in our model. For user U_i, we define $\mathcal{F}(i) = \{f_i^1, f_i^2, ..., f_i^{K(i)}\}$ where f_i^k is a vector of contexts associated with the k^{th} check-in point from U_i (i.e., v_i^k). For simplicity, we refer to each f_i^k as a "context". Formally, we define context as:

DEFINITION 1: ***Context:*** *The context is presented as a vector of categorical variables (e.g. ["sunny","Monday", "evening"]) that is associated with a particular check-in point.*

Take Foursquare as an example. Each user account is considered as a user. Examples of venue category

from check-in venues include "food", "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 of f_i^k can be {"home", "Friday", "cloudy", "football game"}.

Note that some LBSN applications have hierarchical category structures. For example, in Foursquare, "French restaurant" and "Asian restaurant" are subcategories of "Food". "Lyonese Bouchon" and "Southwestern French Restaurant" are subcategories of "French restaurant". In this paper, we only focus on predicting the top-level categories to make the solution general to all types of LSBN applications. However, we do leverage the fine-grained category hierarchy to infer user similarities in Section 4.4.

To formulate the sparse check-in venue prediction problem, we first define a User Check-in Venue Query.

DEFINITION 2: User Check-in Venue Query (UCVQ): a UCVQ asks what is the user's *next check-in venue* given the user's historical check-in venues and the current context. An example of UCVQ would be: "What's the most likely venue Alice would visit from her office in a sunny Friday afternoon when there is a beer festival?"

The objective of the user check-in venue prediction is to answer the query as accurately as possible. To formulate the goal, we first describe the input and output of the problem.

Input: the input to predict each user's next check-in venue include the user's historical check-ins as well as the category and context of each historical check-in. In particular, the inputs for each user include: 1) the *check-in trace vector* $\mathcal{V}$ which is an array of check-in venue IDs; 2) the *check-in category trace vector* $\mathcal{P}$ which is an array of check-in category that corresponds to the check-in venues in $\mathcal{C}$; 3) the *context trace vector* $\mathcal{F}$ which is an array of context entries corresponds to the check-in venues in $\mathcal{V}$. Each context entry is coded as a tuple of the values of the context variables; 4) the context (such as the weather, and day of the week) of the next check-in.

Output: the output is the predicted venue ID as well as the venue category that the user will check in next.

Formally, our goal is to find the venue the user U_i is most likely to visit given the sparse check-in traces and relevant contexts:

$$\arg\max_{v_i^{K(i)+1}} Pr(v_i^{K(i)+1} | f_i^{K(i)+1}, \mathcal{V}, \mathcal{C}, \mathcal{F}), \forall i, U_i \in U \quad (1)$$

where $v_i^{K(i)+1}$ denotes the predicted venue that user U_i would check in, and $f_i^{K(i)+1}$ denotes the contextual information of the check-ins to be predicted.

4 SOLUTION

In this section, we present the CSCVP scheme to address the sparse user check-in venue prediction problem formulated in the previous section.

4.1 Overview of CSCVP Scheme

An overview of the CSCVP system is shown in Figure 1. The CSCVP scheme consists of the following key components: Temporal-Aware Category Prediction (TACP) module, Context-Aware Category Prediction (CACP) module, User Similarity Regulation (USR) module, User Regularity Detection (URD) module, and Check-in Venue Prediction (CVP) module. Considering the importance of category information in sparse user venue prediction, we perform a two-step prediction in CSCVP: 1) we develop new prediction models inspired by NLP techniques to first predict the category of the check-in venues; 2) we develop an active learning based framework to predict the accurate check-in venues by using the category information predicted in the first step. The two-step prediction design is crucial in addressing the data sparsity issue where the prediction space (i.e., number of venues) is very large compared to each individual's check-in data. This 2-step approach allows a significant reduction of candidate venue space for the venue prediction task.

In particular, we first develop a TACP module to predict the category of a user's next check-in venue by exploring the temporal pattern of the user's historical check-in traces. Second, we propose a CACP module to enhance TACP module by incorporating the latent decision context variables of a user and the semantic implication of the context variables. Third, we design a USR module to further alleviate data sparsity issue in the category prediction by considering the similarity between users and the peer's influence on a user's decision process. Fourth, we develop an entropy-based URD module to dynamically decide the pattern regularity of each user's check-in trace, which allows CSVP to smartly decide which category predictor (i.e., TACP or CACP) is more appropriate for a user. Fifth, we integrate the above components together to accomplish the category prediction task using a semi-supervised co-training framework. Finally, we solve the sparse user check-in venue prediction problem by using the category prediction results and an active learning technique, namely Query By Committee (QBC). We discuss the details of these components in the rest of this section.

4.2 Temporal-Aware Category Prediction Module to Explore Dynamic Temporal Dependency

TACP module consists of a novel extension of Ngram model from NLP, Temporal Aware Ngram model (TA-Ngram), and a smoothing procedure.

4.2.1 Basic Ngram Model

Prediction models for categorical data, especially word sequence, are well studied in Natural Language Processing field. Among these models, Ngram based model is one of the most widely adopted solutions [35]. The basic idea of Ngram model is to use the previous N-1 words to predict the current (N^{th}) word, where N is the

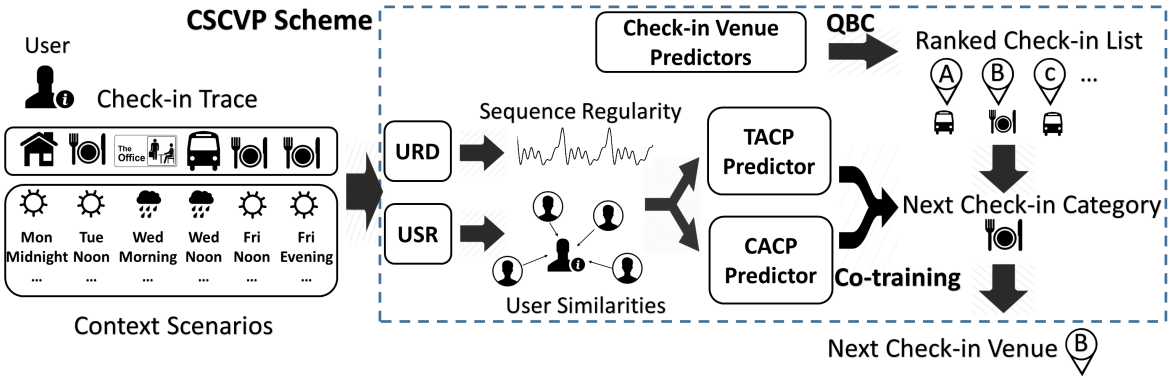


Figure 1: CSCVP Scheme

size of the Ngram model. In the check-in category prediction problem, we map "words" to venue categories and "Ngram" to a sequence of N consecutive venue categories (which is referred to as "sequences" in 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 check-in category using the Ngram model. In particular, the n-th check-in category can be predicted as:

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

where p_i denotes the i^{th} check-in category and p_j^i denotes a sequence of consecutive venue categories $\{p_i, p_{i+1}...p_j\}$ (i.e., p_j^i is an Ngram when $j = i + N - 1$).

4.2.2 A Temporal Adaptive Ngram Model

We observe a few technical challenges of applying the basic Ngram model to solve our check-in category prediction problem. The first challenge comes from the data sparsity issue where the user check-in trace is usually sparse and the time gap between two consecutive check-ins is sometimes large (e.g. several months). Therefore, the dependency assumption between consecutive check-ins made by Ngram might not always hold. Second, the basic Ngram model ignores the freshness of check-ins: the recent check-in categories may be more relevant to the next check-in category compared to the old ones.

To make Ngram model more suitable to our check-in category prediction problem, we develop a Temporal-Aware Ngram model (TA-Ngram) to address the above challenges. In particular, to remove the non-existing dependency between consecutive check-ins, we insert a breaking character (like a "period" in a sentence) between categories p^{i-1} and p^i when the time gap between them is too large. We exclude any category sequence that contains breaking character from the training corpus. The usage of breaking character nicely addresses the invalid temporal dependence between consecutive check-ins with large time gaps, which is caused by the unique data sparsity issue. To capture the "freshness" of the check-in category sequence, we use the following function to adjust the weights on the counts of a sequence in the training corpus. Note that the breaking characters ensure all the generated grams are within a relatively small time interval. For an occurrence of check-in sequence $seq(t)$ at the time interval t, we compute its weight as:

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

where Inv denotes the total number of time intervals of the check-in trace and the α is a tuning parameter. The above function assigns a higher weight to the occurrence of the sequence that happens more recently in the check-in trace. We update the count of each sequence in the training corpus by doing a weighted sum over all of its occurrences.

We further observe that the TA-Ngram may fail to predict the next check-in category accurately if there is a "dominant" check-in category in the user's check-in trace. Here the dominant check-in category refers to the category that a user frequently visits. Consider a check-in trace of a user as "Food, Movie, Food, Work, Food, Nightlife, Food, Travel, Food, Travel, Food", the dominant check-in is "Food", since it is the most frequently visited. Therefore, it is reasonable to predict the next check-in as "Food" since it seems to be the user's habitual check-in behavior. However, using the vanilla Ngram (e.g., bi-gram) model will never predict the next check-in to be "Food" because the Ngram sequence "Food, Food" never appears in the trace, therefore has zero count in the training corpus of Ngram. On the other hand, it is also reasonable to predict the next check-in as "Travel" since the user also exhibits "Food, Travel" pattern. To address this problem, we apply the Witten-Bell smoothing technique used for word prediction in NLP [36]. The Witten-Bell smoothing assigns additional counts to the zero-count Ngram "Food, Food", so that in the next prediction, both "Food" and "Travel" has a chance to be predicted using the Ngram model.

In particular, we update our prediction using TA-Ngram as follows:

$$\begin{aligned} Pr(p_n|p_{n-1}^1) = \lambda(p_{n-1}^1)\frac{c(p_n^{n-N+1})}{c(p_{n-1}^{n-N+1}\bullet)} \\ + (1 - \lambda(p_{n-1}^1))Pr(p_n) \end{aligned} \tag{4}$$

$$\lambda(p_{n-1}^1) = \frac{c(p_{n-1}^{n-N+1}\bullet)}{c(p_{n-1}^{n-N+1}\bullet) + c_{1+}(p_{n-1}^{n-N+1}\bullet)} \quad (5)$$

where $p_j^i\bullet$ denotes a sequence starting with p_j^i and ends with any check-in category including the dominant one. $c_{1+}(p_j^i\bullet)$ is the count of unique sequence starting with $\{p_i, p_{i+1}, ..p_j\}$ in training corpus. λ is a key weighting factor that controls the importance of a dominant category. In particular, the lower value λ is, the higher weight is assigned to the dominant venue category that a user frequently visits.

4.3 Context-Aware Category Prediction Module to Explore Latent User Decision Context

While the TA-Ngram model in the TACP module captures the temporal patterns ("syntactic") of the check-in category trace, it does not consider the contextual information (e.g., weather, time of day, social events) related to each check-in point and its implication to the latent decision process ("semantics") of a user. For example, a context of "Sunday Night" may imply a decision context of "party" for user A but a decision context of "study" for user B, which could lead to different check-in behaviors. Another inherent limitation of the Ngram-based model is the "long-range dependency" where the model fails to explore the correlation between check-ins that are far apart from each other (i.e., beyond of the size of N) or have large time gaps due to the data sparsity issue. In this subsection, we develop a Context-Aware Category Prediction (CACP) module to explore the latent user decision context and further enhance the TACP module through latent semantic analysis. The CACP module leverages the extra context information of each check-in and uses it to identify the behavior pattern by answering the question "given a particular context, where would the user tend to go?". The CACP module is agnostic of the temporal pattern and dependency thus is more robust against the data sparsity issue.

4.3.1 Basics of Probabilistic Latent Semantic Analysis

To address the limitations of TACP module mentioned above, the CACP module adopts a Probabilistic Latent Semantic Analysis (PLSA) based model. The PLSA model fits nicely into our problem because i) it can capture the latent factors (we refer it to the latent decision contexts) that affect users' preferences on venue 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; 3) PLSA explores the semantic correlations of the check-in points to compensate for the 'long-range dependency" limitation of Ngram. While there exist similar latent semantic analysis models such as Latent Dirichlet Allocation (LDA) model [30], we pick PLSA due to its simplicity and good performance in the solving the venue prediction problem [37].

4.3.2 PLSA Based Context-aware Predictor

We first define a set of latent decision context $Z = \{z_1, z_2, ..., z_{|Z|}\}$ to represent a group of hidden factors that lead to the user's decision to visit a check-in category under a specific context. For example, a context of "Friday night" may represent a decision context of "party", which contributes to the user's decision to go to nightlife places. Similarly, a "Monday morning" may represent the decision context of "work", which may contribute to the user's decision to visit public transportation. The idea of PLSA based prediction model is illustrated in Figure 2.

To simplify our notations, we denote a check-in category as p, a context vector as f, and a latent decision context as z, we can compute probability of co-occurrences of p and f:

$$Pr(p, f) = \sum_{z \in Z} Pr(p|z)Pr(z|f)Pr(f), p \in P, f \in F \quad (6)$$

where $Pr(p|z)$ denotes the probability that a user decides to visit category p given a latent decision context z and $Pr(z|f)$ denotes the probability of the latent decision context z given a context f. P and F denote the sets of all distinct venue categories and context vectors for a user.

The likelihood of observing the co-occurrences of p and f in the check-in trace can be written as:

$$\begin{aligned} L(p, f) &= \prod_{p \in P} \prod_{f \in F} Pr(p, f)^{n(p,f)} \\ &= \prod_{p \in P} \prod_{f \in F} (\sum_{z \in Z} Pr(f)Pr(p|z)Pr(z|f))^{n(p,f)} \end{aligned} \quad (7)$$

where $n(p, f)$ is the count of the co-occurrences of p and f in the trace. The estimation parameters are $\theta_{PLSA} = \{Pr(p|z), Pr(z|f)\}$. Note that we assume statistical independence between check-ins in the above formulation. In particular, we intentionally ignored the temporal dependency between the check-in points while only focusing on the conditional dependencies of latent decision context, contextual information, and the venue category. The intuition of this design is to allow the CACP module to capture the long-range dependencies of the check-in categories which cannot be captured by the sequence-based predictive models (e.g., Ngram, Markov, autoregression) that only assume temporal dependency among consecutive check-ins.

We use the EM algorithm to find the optimal estimation of the parameters. Finally, we predict user u's next check-in category as:

$$Pr(p|f) = \sum_{z \in Z} Pr(p|z)Pr(z|f) \quad (8)$$

4.4 User Similarity Regulation Module to Alleviate Data Sparsity

Though the check-in category prediction requires a much smaller prediction space than the actual venue prediction, data sparsity still exists for fresh LBSN users or

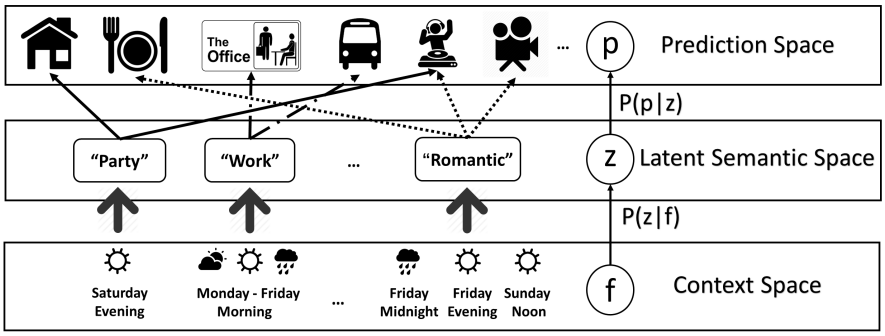


Figure 2: PLSA based Model for Venue Category Prediction

for users who infrequently upload their check-in points to LBSN. To alleviate such sparsity problem, CSCVP explores the similarities between users and the peer's influence on a user's decision process. The intuition is that, while a user's personal check-in trace can be sparse, other similar users' check-in behavior can provide extra evidence for venue prediction. Since many real-world data traces did not include a social graph to generate the similarity matrices between users [22], [38], we infer the user similarity using the following metrics:

DEFINITION 3: Pattern Similarity $sim_{Pat}(i,j)$: it measures how similar the check-in pattern of user j is compared to user U_i.

DEFINITION 4: Context Similarity $sim_{Con}(i,j)$: it measures how similar the check-in behavior of user j is (under a specific context) compared to user U_i.

To measure the pattern similarity, we divide check-ins of user U_i into T time intervals. If there is no check-in during an interval, we use a special category as the placeholder. Let T_i^t denote the check-ins at the t^{th} interval for user U_i, we have:

$$sim_{Pat}(i,j) = \sum_{t=1}^{T} \delta(T_i^t, T_j^t)/T \tag{9}$$

where δ is a hierarchical similarity score function defined as follows.

$$\delta(T_i^t, T_j^t) = \begin{cases} a_k, T_i^t \text{ and } T_j^t \text{ match level-k subcategory} \\ 0\ , T_i^t \text{ and } T_j^t \text{ have different root categories} \end{cases} \tag{10}$$

where $0 < a_k \leq 1$ is the weight for category match at level-k subcategory ($k = 1$ for a root category). In this work, we only consider two levels of categories (more details are explained in the next section) and we assign $a_2 > a_1$ since a more refined match indicates a higher similarity between the users' traces. In particular, we define a "match" as $T_i^t \cap T_j^t \neq \emptyset$ given the fact that a user can visit multiple venue categories at each time interval.

To measure the context similarity, we define CT(i,f) as all the check-ins for user U_i under a specific context f. In particular, we find a user's most preferred category under every possible context and compute the hierarchical preference similarity scores of these categories as follows.

$$sim_{Con}(i,j) = \frac{\sum_{f \in F(i)} \delta(fq\{CT(i,f)\}, fq\{CT(j,f)\})}{size\ of\ F(i)} \tag{11}$$

where $fq\{CT(i,f)\}$ denotes the most frequently visited check-in category in $CT(i,f)$ and $F(i)$ denotes all possible context scenarios for user U_i. The intuition here is that if two users happen to prefer the same check-in category under a specific context, they are more likely to visit the same check-in category next time under the same context.

We use the above similarity metrics to regulate the prediction results of TACP and CACP modules to alleviate data sparsity issue. In particular, let $TA(i,N)$ and $CA(i,K)$ denote the output of the TACP and CACP modules for user U_i respectively. N is the size of TACP and K is the number of topics in CACP. We compute the regulated result as:

$$\begin{aligned} TA'(i,N) &= TA(i,N) + \sum_{j \in U, j \neq i} TA(j,N) * sim_{Pat}(i,j) \\ CA'(i,K) &= CA(i,K) + \sum_{j \in U, j \neq i} CA(j,K) * sim_{Con}(i,j) \end{aligned} \tag{12}$$

The results are normalized to meet the constraint $\sum_{p \in P} Pr(p|f) = 1$.

4.5 User Regularity Detection Module and Prediction Integration

The TACP module works better for the check-in traces with strong regularity (e.g., a user periodically visits a set of venue categories) and the CACP module 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 define a "regular" check-in trace as a trace that has a strong periodical pattern (e.g., "Food, Movie, Food, Movie, Food Movie..."). To identify the "regularity" of a check-in trace, we instead focus on its dual problem: find the "randomness" or "irregularity" of the trace, which can be effectively solved by using the entropy-based approaches from information theory. In general, a sequence with a lower entropy is more regular [39].

However, one issue of using the entropy-based solutions is that they ignore the position of each category in a sequence. For example, a sequence of "Food, Movie, Food, Movie, Food, Movie" yields the same entropy score as "Food, Food, Movie, Food, Movie, Movie" while the former sequence clearly has a stronger regularity. We also observe that a bi-gram model is very suitable for prediction of the first sequence since the category "Movie" always appears after "Food".

We develop a URD module to jointly detect the regularity of a check-in trace and identify the optimal size N for the TACP module. In particular, we first define a sequence mapping function $\mathrm{Map}(ws, seq)$ to convert the check-in category 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 TACP, we calculate the window size for user U_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.

$$\begin{aligned} N = ws_i &= \arg\min_{n} H(\mathrm{Map}(n, \mathcal{P}(i))) \\ H(seq) &= -\sum_{pos_j \in seq} Pr(pos_j) * \log(Pr(pos_j)) \end{aligned} \tag{13}$$

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.

We now discuss how to integrate all components into a holistic check-in category prediction scheme. In short, we first use TACP and CACP to predict the next check-in category independently and then integrate the results using a co-training framework.

Co-training is a semi-supervised learning technique that requires two weak predictors where each predictor represents a different *view* of the check-in trace data [40]. By combining the complementary information and features from different views, co-training can often generate better prediction results compared to a single predictor alone [40], [41]. In our model, we use TACP and CACP as two predictors where the TACP explores the temporal patterns and the CACP explores the context/semantic features. We present our final scheme using the co-training framework in Algorithm 1. The intuition is that we use the TACP predictor when a user's check-in trace is identified as regular (NH_i is small enough) and use CACP 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.

Algorithm 1 Co-training Learning for CSCVP

Input: Check-in trace of a users $\mathcal{P}(i)$, divided into labeled (training) set S_l and unlabeled set (test) S_u of venue categories. A parameter η controls maximum number of iterations. User id i.
Output Prediction results of all unlabeled venue categories.

$iter \leftarrow 0$
while $iter = \eta$ or $S_u = \emptyset$ **do**
  Train TACP and CACP using S_l.
  Identify optimal N based on Equation (13).
  Predict unlabeled venue categories in S_u with TACP and CACP separately based on Equation (12).
  for $p \in P$ **do**
    Find the set of check-ins $\in S_t$ that predicted as p with significant confidence ($probability \geq 0.8$), move the set into S_l. Repeat this step for both predictors.
  end for
  $iter \leftarrow iter + 1$
end while
if $NH_i \leq thres_l$ **then**
  Return $TA'(i, N)$
else if $NH_i \geq thres_h$ **then**
  Return $CA'(i, K)$
else
  Return $CA'(i, K) \times TA'(i, N), \forall i, U_i \in U$
end if

4.6 User Check-in Venue Prediction

Finally, we show how the CSCVP scheme uses the check-in category prediction results obtained from the above modules to predict the actual check-in venues of LBSN users. In particular, we adopt an active learning technique, Query by Committee (QBC) [42], to predict the exact next venue a user will visit by integrating the category prediction results from CSCVP and the venue prediction results from a set of state-of-the-art prediction algorithms. The key idea of QBC is to form a committee of "experts" (an expert refers to a check-in venue prediction algorithm in our case) and select a prediction result that is with the least disagreement (variance) with all experts. We use four recent venue prediction algorithms as committee members - ST-LDA [15], STPMF [12], CIKM13 [10], and GeoCF [8]. The details of these algorithms are discussed in Section 5.2.

Let us assume a check-in venue prediction scheme outputs a ranking list of the predicted venues as $RL^a = \{Venue_1^a, Venue_2^a, ..., Venue_X^a\}$ where RL^a denotes the a^{th} algorithm and $Venue_x^a$ denotes the top x^{th} predicted venue by the a^{th} algorithm. The list is ranked by the likelihood (from high to low) of a venue that a user will visit next. A committee's opinions are the outputs of all experts, namely $RL^1, RL^2, ..., RL^a, RL^A$, where A is the size of the committee. Intuitively, if the predicted next check-in is disagreed to a great extent by committee members, then the quality of the prediction is low. To derive such disagreement, we compute a Disagreement Index (DI) of each venue V_x as:

$$DI(V_x) = \sum_{1 \leq a \leq A} (\mathrm{Rank}^a(V_x) - 1), 1 \leq x \leq X \tag{14}$$

where $\mathrm{Rank}^a(V_x)$ is the ranking position of venue V_x from the a^{th} algorithm. This equation defines the difference of rankings from all experts if V_x were elected as

the predicted venue (i.e., V_x is ranked first by CSCVP). For example, if V_x indeed ranks first in all committee member's ranking lists, the DI equals zero. We rank all venues based on the DI index from low to high. The ranked list is denoted as RL^{QBC}.

Finally, we use the category prediction result to regulate the ranking score of each venue based on the following equation:

$$\mathrm{Rank}^{\mathrm{QBC}}(V_x)' = (1 - Pr(p|f)_x) \times \mathrm{Rank}^{\mathrm{QBC}}(V_x) \quad (15)$$

where $\mathrm{Rank}^{\mathrm{QBC}}(V_x)$ is the original ranking position of venue V_x by QBC. $Pr(p|f)_x$ is the probability of the predicted check-in category of V_x from CSCVP. We re-rank the check-in venue predictions based on $\mathrm{Rank}^{\mathrm{QBC}}(V_x)'$ scores (from low to high) so the predicted venue that matches the category prediction results is more likely to appear on the top of the re-ranked list.

In summary, the benefit of using QBC approach is that it eliminates the needs to re-invent the wheel (i.e., developing a completely new check-in venue prediction scheme from scratch) but rather provides a generic framework to leverage a pyramid of well-recognized, state-of-the-art venue prediction solutions regulated by our CSCVP scheme.

5 Evaluation

In this section, we evaluate the performance of the CSCVP scheme and compare it to state-of-the-art baselines on three real-world LBSN data traces. The results show that the CSCVP scheme outperforms all baselines in terms of accurately predicting the venue a LBSN user would check in on all data traces.

5.1 Data Collection and Pre-Processing

In the evaluation, we use three real-world Foursquare data traces[1] collected from New York City (NYC), Tokyo (TKY), and Paris (PAR) between Apr. 3rd 2012 and Sep. 16 2013 [38]. The statistics of the three data traces are summarized in Table 1. We also plot the Empirical Cumulative Distribution Function (ECDF) with regard to the number of check-ins per user in Figure 3. We choose to use log scale for the number of check-ins due to the long tail distribution (e.g., few users have over 500 check-ins). It is shown that the number of check-ins per user is quite diverse. We observe the data sparsity issue in all data traces. In particular, the PAR data trace has a much smaller number of check-ins per user than other traces, which provides us an excellent real-world scenario to evaluate the robustness of the CSCVP scheme. The diversified locations of the data traces also help us to evaluate the performance of our scheme over different cultural and regional settings.

For the CSCVP scheme, we select three context variables for our CACP predictor in the evaluation. We define and process these context variables as follows:

1. https://sites.google.com/site/yangdingqi/home/foursquare-dataset

Data Trace	NYC	Tokyo	Paris
# of Users	1,083	2,293	6,903
# of Venues	38,333	61,858	19,837
# of Total Check-ins	227,428	573,703	111,325
Check-ins per User	210.0	250.2	16.1
Check-ins per Venue	5.9	9.3	5.6

Table 1: Data Trace Statistics

- **Weather:** We first collect historical daily weather information from Weather Underground [2] that covers the data collection period of each data trace. We then build a Groovy script to automatically label the weather condition of each check-in point based on its geolocation and timestamp.
- **Day of the Week:** We label each check-in point as happening on either "weekday" or "weekend" based on the date of the check-in.
- **Time of the Day:** We label each check-in point 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.

Several previous works explored the social graph context [8], [32], [43]. However, the social network information is not available in our data traces. So we focus on the similarity between a user's visiting behavior and context. The "spatial locality" has also been explored as a context feature based on the assumption that users tend to visit places that are of close proximity [8], [23]. However, this assumption does not hold on LBSN data traces we collected: a significant amount of users do not upload the complete check-in trace online (e.g., due to privacy concerns) and the consecutive check-in points in the collected trace are not necessarily of close proximity.

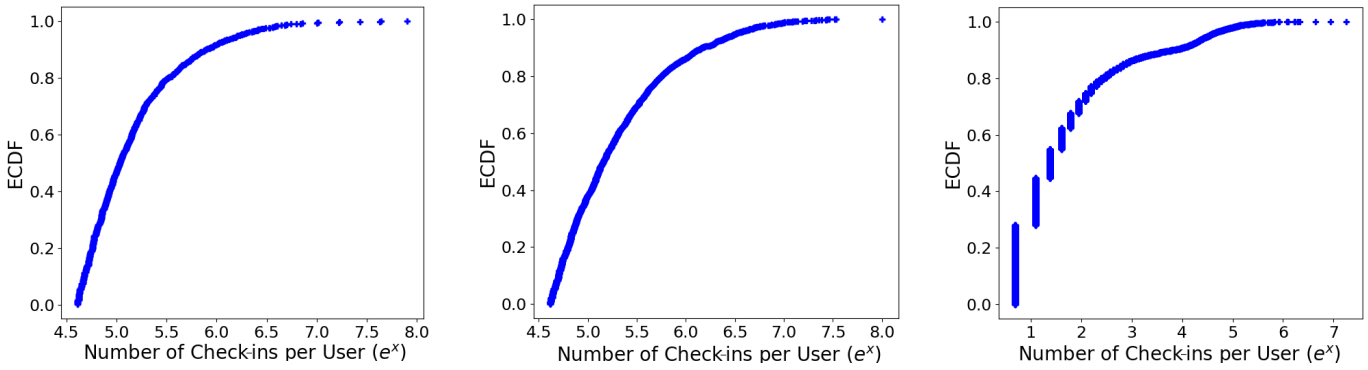
Finally, we use the nine categories (i.e. Arts & Entertainment, College & University, Food, Outdoors, Nightlife, Professional & Other Places, Residence, Shop & Service, Travel & Transport) defined by the official Foursquare developer documentation[3] as the candidate classes for our venue category prediction task.

5.2 Baseline Methods

We choose the following representative check-in venue prediction schemes from the literature as our baselines. Note that the first 3 baselines are designed for category prediction and the remaining 9 baselines are designed for the check-in venue prediction. For the venue prediction baselines, we use the corresponding category of the predicted venue as the results of the category prediction experiments.

- **CAP-CP:** Our previous venue category prediction scheme that ignores user's similarity and peer influence in its prediction. It leverages a linear combination of Ngram and PLSA predictors [31].

2. www.wunderground.com/history/
3. developer.foursquare.com/categorytree



(a) NYC Check-in Distribution (b) TKY Check-in Distribution (c) PAR Check-in Distribution

Figure 3: Check-in Distribution for All Data Traces

- **TA-Ngram:** Our proposed TA-Ngram model with N varying from two (bi-gram) to six.
- **PLSA:** Our proposed PLSA model with the topic number varying from one to ten.
- **Popularity-General:** It ranks the set of venues that a user visits in terms of general popularity (measured as the total number of visitors) [44].
- **Popularity-Time:** It ranks the set of venues that a user visits in terms of popularity of a venue at the hour of the week where a check-in takes place [44].
- **Popularity-Distance:** It ranks the set of venues that a user visits in terms of distance from the user's most frequently visited location (also referred to as home location in the literature) [44].
- **Frequency:** It ranks the set of venues that a user visits in terms of the user's most frequently visited location. We assign probability of being in a certain venue equal to the normalized frequency of visit.
- **ST-LDA:** A latent probabilistic generative model that learns region-dependent personal interest for venue prediction [15].
- **STPMF:** A Spatial-Temporal Probabilistic Matrix Factorization model that explores a user's geographical preference and other general interests in venue prediction [12].
- **CIKM13:** A category-aware venue prediction model that exploits the transition patterns of users' preference over location categories in its prediction [10].
- **GeoCF:** A collaborative filtering based venue prediction scheme that combines the user preference and geographical influence in its prediction [8].
- **QBC:** An active learning technique that aggregates the prediction results from STPMF, CIKM13, GeoCF and ST-LDA schemes for the venue prediction [42].

There are two main methods to evaluate the check-in prediction problems: 1) randomly mask off some check-ins in each user's trace and predict them [15], [24]; 2) split the data trace in chronological order and only mask off the most recent check-ins [10], [45]. In our experiment, we adopted the latter method because in the real-world scenario, it is not possible nor interesting to "look back" the data trace and predict earlier check-ins. Similar to [45], we use each user's first 80% check-in records in chronological order to create the training set examples and then use the remaining 20% as the test set.

The parameters of the baselines are fine-tuned by performing 10-fold cross validation using the data from the training set. In particular, each time we use 90% of the training data to train a model with a given setting of parameters. Then the rest 10% of data is used to validate the performance of current parameter setting based on the F-1 score. Note that exhausting all parameter settings can be computationally expensive (i.e., the complexity grows exponentially as the number of parameters increases). We adopt an efficient model tuning technique called Grid Search [46] to identify the best-performing parameter setting. We set the time gap as one 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 data traces. We also set $\alpha = 5.6, 4.2, 5.0$ for NYC, TKY and PAR data traces, respectively. For a fair comparison, we choose the parameter setup that yields the best results for the baselines that have tunable parameters.

5.3 Evaluation on Real World Data Traces

5.3.1 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, 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 venue predictions or recommendations [9]. In particular, Precision@K defines the ratio of successfully predicted labels (i.e., venues or venue categories) to the top K predictions, and Recall@K defines the ratio of successfully predicted labels to the total number of labels to be predicted (i.e., K).

Schemes	Accuracy			Precision			Recall		
	NYC	TKY	PAR	NYC	TKY	PAR	NYC	TKY	PAR
CSCVP	**0.5301**	**0.6671**	**0.4583**	**0.2837**	**0.3004**	**0.2588**	**0.8511**	**0.9012**	**0.7764**
CAP-CP	0.4915	0.6275	0.4182	0.2599	0.2761	0.2305	0.7796	0.8283	0.6915
TA-Ngram	0.3771	0.5169	0.3013	0.2342	0.2893	0.2084	0.7026	0.8679	0.6553
PLSA	0.3267	0.4392	0.3223	0.1744	0.2580	0.2166	0.5232	0.7741	0.6497
Popularity-General	0.2301	0.4652	0.2926	0.1698	0.2654	0.2144	0.5092	0.7962	0.6432
Popularity-Time	0.3074	0.5152	0.3381	0.2077	0.2675	0.1987	0.6231	0.8026	0.5961
Popularity-Distance	0.1590	0.3538	0.2871	0.1254	0.1972	0.1776	0.3762	0.5916	0.5327
Frequency	0.3912	0.5498	0.4014	0.2388	0.2718	0.2277	0.7164	0.8154	0.6831
QBC	0.4577	0.5803	0.3919	0.2539	0.2815	0.2273	0.7618	0.8445	0.6682
ST-LDA	0.4372	0.5217	0.3315	0.2324	0.2743	0.1971	0.6973	0.8229	0.5912
STPMF	0.4397	0.5116	0.3529	0.2209	0.2775	0.2091	0.6228	0.8325	0.6274
CIKM13	0.4173	0.5023	0.3720	0.2383	0.2668	0.2173	0.7150	0.8003	0.6519
GeoCF	0.3123	0.4983	0.3189	0.1970	0.2473	0.2069	0.5907	0.7419	0.6206

Table 2: Category Prediction Performance of All Schemes

5.3.2 Evaluation Results on Category Prediction

Category prediction is a critical component of the CSCVP scheme and the ultimate venue prediction requires an accurate prediction of the venue category in the first place. In the first set of experiments, we evaluate the performance of the category prediction component of CSCVP in terms of accuracy, precision, recall, and robustness.

Table 2 shows the performance results of all compared schemes on three different data traces. We observe that our CSCVP scheme significantly improves the performance of our previous CAP-CP scheme in all three metrics. We attribute this performance gain to i) the user similarity regulation module where peer similarities and influence are used to compensate for sparse check-in data; ii) the co-training framework that is more effective than the naive linear combination used by CAP-CP to integrate results from two different predictors. In addition, our scheme also significantly outperforms other baselines on all evaluation metrics across different data traces. In particular, the CSCVP scheme improves the category prediction accuracy by 9.29% in NYC data trace, 14.54% in TKY data trace, and 12.68% in PAR data trace compared to the best-performing baseline, respectively. 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 latent user decision context in their prediction model. We observe that the heuristic baselines based on the popularity (i.e., Popularity-General, Popularity-Time, and Popularity-Distance baselines) or frequency (i.e., Frequency), did not perform well. Especially, Popularity-General and Popularity-Distance performed quite poorly in all metrics. This is mainly due to the fact that 1) users have diverse check-in behaviors which may not necessarily follow the preferences of all users in general; 2) we observe that users do not necessarily check in to venues that are closest to their home locations. In fact, we found the average distance is 6.89 km between a check-in point and the user's home location. We show the detailed distance distribution in Figure 4. The recall and precision are evaluated based on the top 3 predicted venues (i.e., $K = 3$) which are chosen considering the prediction space of 9 categories on Foursquare. The non-

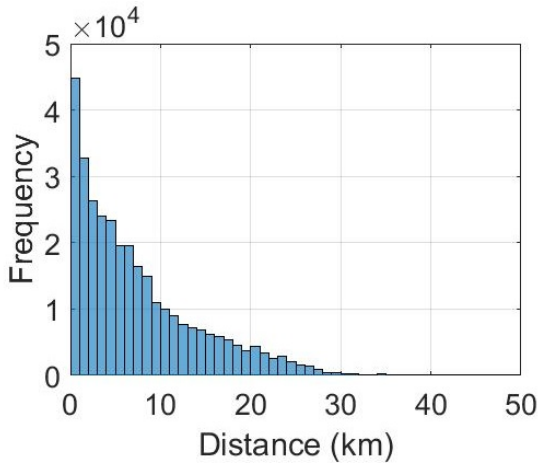


Figure 4: Venue Distance to User's Home Location

trivial performance gains of CSCVP on both precision and recall demonstrate that our scheme can predict the next check-in category with fewer guesses than other baselines. We also observe that all schemes perform better in the TKY data trace than other data traces. We attribute this result to the fact that the check-in traces of Tokyo users tend to have a stronger temporal pattern than users in the other two cities. In fact, the average entropy of TKY users (1.796) is much lower than the other two cities' (2.351 for NYC and 2.241 for PAR). The results also show that our scheme has consistent better performance than the baselines when data is sparse (i.e., PAR data trace). This is achieved by the user similarity regulation module in CSCVP scheme that predicts user's venue category by exploring the peer influence when his/her trace is sparse.

To more closely investigate the performance gain achieved by CSCVP scheme, we consider not only whether the prediction is correct but also which categories are confused with each other. Table 3 shows confusion matrices for CSCVP and QBC, one of the best performing baselines from the previous experiment. We observe that CSCVP outperforms the QBC incorrectly predicting the majority of the venue categories and

Confusion Matrix: CSCVP

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: QBC

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 3: Confusion Matrix for CSCVP and QBC

minimizing possible confusions (i.e., large numbers of diagonal elements and small numbers in others). For example, we observe that QBC has trouble recognizing the Shop category (i.e., 28%) and confuses it with Food. This is because that the users' traces on these two categories often mingle together, making it hard for the QBC to make the right prediction. In contrast, CSCVP has a high accuracy for the Shop category (i.e., 59%) thanks to the latent user decision context incorporation in CACP module (e.g., users are more likely to go to shopping venues on "weekends" or "sunny/cloudy days").

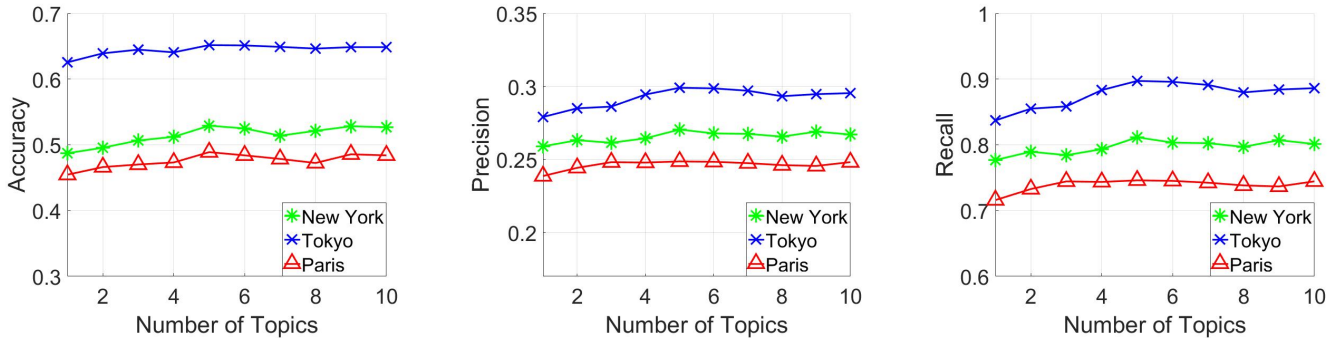
We also study the robustness of the CSCVP scheme by tuning the parameters of our model. One key parameter we use is the number of topics in the PLSA based predictor. This parameter directly controls the possible number of latent decision context variables we consider in our model. The results are shown in Figure 5. We observe that the performance of CSCVP is relatively stable when the number of topics changes.

5.3.3 Evaluation Results on Venue Prediction

Finally, we demonstrate that the CSCVP scheme can significantly outperform the state-of-the-art baselines in solving the actual sparse user check-in venue prediction problem. For the popularity and frequency based baselines, we picked the two best-performing ones: Popularity-Time ("Popularity" for short) and Frequency to evaluate the performance gain. We continue to use Precision@K and Recall@K metrics. The results are shown in Figure 6 and Figure 7. We observe that our CSCVP achieves non-trivial performance gains compared to all baselines. We attribute such performance gains to the accurate category predictions of CSCVP as shown in the previous subsection. We also note that the performance gain achieved by CSCVP scheme is more obvious in the PAR data trace where data sparsity is very significant (an average of 16.1 check-in points per user). This is due to the fact that our scheme explicitly addresses data sparsity issue by leveraging both the category information and the similarity and peer influence between users on LBSN. We further present the 90% confidence interval for all compared schemes. The results are shown in Tables 4 and 5. We observe that the CSCVP scheme has one of the tightest confidence bounds, which demonstrate that our scheme can accurately predict next check-in venue with a low uncertainty.

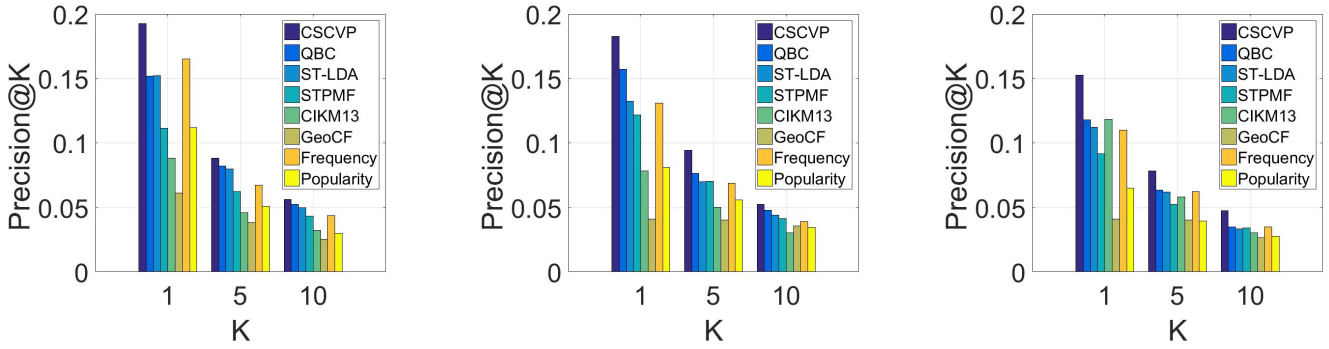
6 Conclusion

This paper develops a CSCVP scheme to address three fundamental challenges in solving the sparse user check-in venue prediction problem on LBSN: *data sparsity*, *latent decision context incorporation* and *dynamic and non-deterministic temporal dependency*. In particular, the CSCVP scheme first predicts the categories of venues to address data sparsity challenge by significantly reducing the prediction space. We develop a PLSA based model that explores the latent semantic settings in the user's visit decision process to address the latent decision context incorporation challenge. We also develop a TA-Ngram model that captures the dynamic dependency between check-in points from the user's trace to address the dynamic and non-deterministic temporal dependency challenge. The predicted category information is used together with an active learning based solution (QBC) to predict the next check-in venue of a user with sparse check-in trace. The evaluation results on three real-world data traces show that the CSCVP scheme can significantly outperform the performance of current check-in venue prediction schemes. The results of this paper are important because they offer a novel analytical framework inspired by NLP techniques to effectively solve the sparse user check-in venue prediction problem in particular and the sparse context-aware location-based prediction problems in general on LBSN.



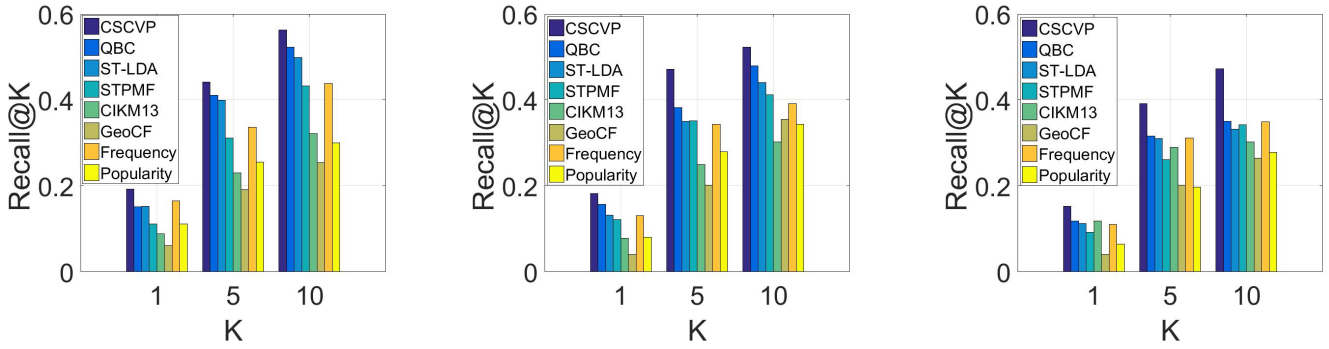
(a) Accuracy by # of Topics (b) Precision by # of Topics (c) Recall by # of Topics

Figure 5: Category Prediction Performance by Tuning Number of Topics



(a) NYC Precision@K (b) TKY Precision@K (c) PAR Precision@K

Figure 6: Check-in Venue Prediction Precision@K of All Compared Schemes



(a) NYC Recall@K (b) TKY Recall@K (c) PAR Recall@K

Figure 7: Check-in Venue Prediction Recall@K of All Compared Schemes

Schemes	CSCVP	QBC	ST-LDA	STPMF	CIKM13	GeoCF	Frequency	Popularity
Confidence Interval for NYC	± 0.0131	±0.0154	±0.0172	±0.0123	±0.0231	±0.0270	±0.0177	±0.0162
Confidence Interval for TKY	± 0.00486	±0.00753	±0.00591	±0.00439	±0.00583	±0.00972	±0.00742	±0.00699
Confidence Interval for PAR	± 0.0346	±0.0622	±0.0795	±0.0415	±0.0677	±0.0553	±0.0816	±0.0621

Table 4: Confidence Interval (Precision) of All Schemes

Schemes	CSCVP	QBC	ST-LDA	STPMF	CIKM13	GeoCF	Frequency	Popularity
Confidence Interval for NYC	± 0.0226	±0.0267	±0.298	±0.0213	±0.0400	±0.0468	±0.0306	±0.0280
Confidence Interval for TKY	± 0.00842	±0.0130	±0.0102	±0.00760	±0.0101	±0.0168	±0.0128	±0.0121
Confidence Interval for PAR	± 0.0599	±0.107	±0.137	±0.0718	±0.117	±0.0958	±0.141	±0.107

Table 5: Confidence Interval (Recall) of All Schemes

ACKNOWLEDGMENT

This research is supported in part by the National Science Foundation under Grant No. CNS-1845639, CNS-1831669, CBET-1637251, CNS-1566465 and IIS-1447795, Army Research Office under Grant W911NF-17-1-0409, Google 2017 Faculty Research Award. 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] D. Wang, B. K. Szymanski, T. Abdelzaher, H. Ji, and L. Kaplan, "The age of social sensing," *Computer*, vol. 52, no. 1, pp. 36–45, 2019.

[2] H. Huang, G. Gartner, J. M. Krisp, M. Raubal, and N. Van de Weghe, "Location based services: ongoing evolution and research agenda," *Journal of Location Based Services*, vol. 12, no. 2, pp. 63–93, 2018.

[3] D. Wang, M. T. Amin, S. Li, T. Abdelzaher, L. Kaplan, S. Gu, C. Pan, H. Liu, C. C. Aggarwal, R. Ganti *et al.*, "Using humans as sensors: an estimation-theoretic perspective," in *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. IEEE, 2014, pp. 35–46.

[4] C. Huang, D. Wang, S. Zhu, and D. Y. Zhang, "Towards unsupervised home location inference from online social media," in *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE, 2016, pp. 676–685.

[5] 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.

[6] H. Gao, J. Tang, and H. Liu, "gscorr: modeling geo-social correlations for new check-ins on location-based social networks," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 1582–1586.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] H. Gao, J. Tang, X. Hu, and H. Liu, "Content-aware point of interest recommendation on location-based social networks," in *AAAI*. AAAI, 2015, pp. 1721–1727.

[12] H. Li, R. Hong, Z. Wu, and Y. Ge, "A spatial-temporal probabilistic matrix factorization model for point-of-interest recommendation," in *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 2016, pp. 117–125.

[13] X. Chen, Y. Zeng, G. Cong, S. Qin, Y. Xiang, and Y. Dai, "On information coverage for location category based point-of-interest recommendation." in *AAAI*, 2015, pp. 37–43.

[14] T. Horozov, N. Narasimhan, and V. Vasudevan, "Using location for personalized poi recommendations in mobile environments," in *Applications and the internet, 2006. SAINT 2006. International symposium on*. IEEE, 2006, pp. 6–pp.

[15] 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.

[16] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura, "Travel route recommendation using geotags in photo sharing sites," in *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 2010, pp. 579–588.

[17] M. Xue, L. Yang, K. W. Ross, and H. Qian, "Characterizing user behaviors in location-based find-and-flirt services: anonymity and demographics," *Peer-to-Peer Networking and Applications*, vol. 10, no. 2, pp. 1–11, 2016.

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

[19] L. Zhao, Y. Lu, and S. Gupta, "Disclosure intention of location-related information in location-based social network services," *International Journal of Electronic Commerce*, vol. 16, no. 4, pp. 53–90, 2012.

[20] 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.

[21] Y. Zhang, Y. Lu, D. Zhang, L. Shang, and D. Wang, "Risksens: A multi-view learning approach to identifying risky traffic locations in intelligent transportation systems using social and remote sensing," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 1544–1553.

[22] S. Jiang, X. Qian, J. Shen, Y. Fu, and T. Mei, "Author topic model-based collaborative filtering for personalized poi recommendations," *IEEE transactions on multimedia*, vol. 17, no. 6, pp. 907–918, 2015.

[23] 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.

[24] 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.

[25] B. Liu, H. Xiong, S. Papadimitriou, Y. Fu, and Z. Yao, "A general geographical probabilistic factor model for point of interest recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 5, pp. 1167–1179, 2015.

[26] 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.

[27] D. Y. Zhang, Y. Zhang, Q. Li, N. Vance, and D. Wang, "Robust state prediction with incomplete and noisy measurements in collaborative sensing," in *2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2018, pp. 460–468.

[28] 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.

[29] 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.

[30] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[31] D. Y. Zhang, D. Wang, H. Zheng, X. Mu, Q. Li, and Y. Zhang, "Large-scale point-of-interest category prediction using natural language processing models," in *2017 IEEE International Conference on Big Data (IEEE BigData 2017)*. IEEE, 2017.

[32] H. Li, Y. Ge, R. Hong, and H. Zhu, "Point-of-interest recommendations: Learning potential check-ins from friends." in *KDD*, 2016, pp. 975–984.

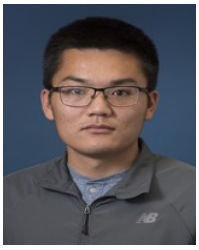
[33] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang, "Towards mobile intelligence: Learning from gps history data for collaborative recommendation," *Artificial Intelligence*, vol. 184, pp. 17–37, 2012.

[34] J. Sang, T. Mei, J.-T. Sun, C. Xu, and S. Li, "Probabilistic sequential pois recommendation via check-in data," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. ACM, 2012, pp. 402–405.

[35] W. B. Cavnar, J. M. Trenkle *et al.*, "N-gram-based text categorization," *Ann Arbor MI*, vol. 48113, no. 2, pp. 161–175, 1994.

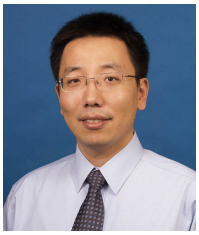
[36] I. H. Witten, A. Moffat, and T. C. Bell, *Managing gigabytes: compressing and indexing documents and images*. Morgan Kaufmann, 1999.
[37] Y. Lu, Q. Mei, and C. Zhai, "Investigating task performance of probabilistic topic models: an empirical study of plsa and lda," *Information Retrieval*, vol. 14, no. 2, pp. 178–203, 2011.
[38] 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.
[39] A. Rényi *et al.*, "On measures of entropy and information," in *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, vol. 1, 1961, pp. 547–561.
[40] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*. ACM, 1998, pp. 92–100.
[41] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: When urban air quality inference meets big data," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 1436–1444.
[42] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992, pp. 287–294.
[43] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, "On truth discovery in social sensing: A maximum likelihood estimation approach," in *Proc. ACM/IEEE 11th Int Information Processing in Sensor Networks (IPSN) Conf*, Apr. 2012, pp. 233–244.
[44] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo, "Mining user mobility features for next place prediction in location-based services," in *2012 IEEE 12th international conference on data mining*. IEEE, 2012, pp. 1038–1043.
[45] M. Hang, I. Pytlarz, and J. Neville, "Exploring student check-in behavior for improved point-of-interest prediction," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 321–330.
[46] P. Lerman, "Fitting segmented regression models by grid search," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 29, no. 1, pp. 77–84, 1980.

Daniel (Yue) Zhang is a PhD student in the Department of Computer Science and Engineering at the University of Notre Dame. He received his M.S. degree from Purdue University, West Lafayette, Indiana, USA, in 2012 and a B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 2008. His research interests include human-centric computing, social sensing based edge computing ecosystem, truth analysis on social media, and Cyber-Physical Systems. He is a student member of IEEE.



Yang Zhang is a PhD student in the Department of Computer Science and Engineering at the University of Notre Dame. His research interests include social sensing, machine learning, edge computing, federated learning, human-cyber-physical systems, human-AI hybrid system, and large-scale real-time distributed systems. He is a student member of IEEE.

Qi Li is a PhD student in the Department of Computer Science and Engineering at the University of Notre Dame. She received her M.A. degree in Statistics at University of Missouri, in 2015. Her research focuses on network science, graph mining, dynamic networks, biological networks, social networks, and machine learning. She is a student member of IEEE.



Dong Wang received his Ph.D. in Computer Science from University of Illinois at Urbana Champaign (UIUC). He is now an assistant professor in the Department of Computer Science and Engineering at the University of Notre Dame. Dr. Wang's research interests lie in the area of social sensing, human-cyber-physical computing, edge computing, and smart cities applications. He received the NSF CAREER Award, Google Faculty Research Award, Army Research Office Young Investigator Program (YIP) Award, Wing-Kai Cheng Fellowship from the University of Illinois and the Best Paper Award of IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS). He is a member of IEEE and ACM.