Place-Type Detection in Location-Based Social Networks

Mohammed Hasanuzzaman
ADAPT Centre
School of Computing, Dublin City University
Dublin 9, Ireland

Andy Way
ADAPT Centre
School of Computing, Dublin City University
Dublin 9, Ireland

ABSTRACT

While most prior studies in Location-Based Social Networks (LSBNs) have mainly centered around areas such as Point-of-Interest (POI) recommendation and place tag annotation, there exists no works looking at the problem of associating place-type to venues in LBSNs. Determining the type of places in location-based social networks may contribute to the success of various downstream tasks such as Point-of-Interest recommendation, location search, automatic place name database creation, and data cleaning.

In this paper, we propose a multi-objective ensemble learning framework that (i) allows the accurate tagging of places into one of the three categories: ***public, private,*** or ***virtual***, and (ii) identifying a set of solutions thus offering a wide range of possible applications. Based on the check-in records, we compute two types of place features from (i) specific patterns of individual places and (ii) latent relatedness among similar places. The features extracted from specific patterns (SP) are derived from all check-ins at a specific place. The features from latent relatedness (LR) are computed by building a graph of related places where similar types of places are connected by virtual edges. We conduct an experimental study based on a dataset of over 2.7M check-in records collected by crawling Foursquare-tagged tweets from Twitter. Experimental results demonstrate the effectiveness of our approach to this new problem and show the strength of taking various methods into account in feature extraction. Moreover, we demonstrate how place type tagging can be beneficial for place name recommendation services.

CCS CONCEPTS

•**Information systems →Information systems applications;** *Location based services;*

KEYWORDS

Location-Based Social Networks; Place-type tagging; POI recommendation

1 INTRODUCTION

Recently, with the rapid development of GPS-enabled smart phones and Web 2.0 technologies, location-based social networks (LBSNs) have become very popular. Typical examples of LBSNs are Foursquare,[1] Facebook Places,[2] Yelp,[3] BrightKite, [4] and Gowalla,[5] etc.. In LBSNs, users can share their locations (e.g., tourist attractions, shops, cinemas, restaurants etc.) via check-in facilities, write reviews, connect with their friends, and upload photos among others.

Location or venue is one of the main concept in LBSNs, and the number of venues in LBSNs is growing continuously. For example, Foursquare had more than 10 million registered users with 1 billion check-ins in September 2011, and by April 2012 the number of check-ins doubled [19]. In LBSNs, a venue can be business, physical location, or virtual location. LBSNs allow registered users to explicitly record their presence at a venue. Users can choose to display their check-in information on their connected friends' Foursquare sites, and post the check-ins on their Twitter or Facebook accounts. Most of the LSBN services allow users to create new venues using various methods,[6] especially when they unable to find their current place during their check-in process. Apart from that, LBSNs users can add "tags" to venues or leave "tips" to venues, which are crucial for assisting users in searching and exploring new places as well as for developing recommendation services [1, 14, 36].

To support various business purposes, most LBSNs services grant users unique opportunities by allowing them to freely create venues, add tags, and leave tips. Although this represents incredible and unique business opportunities, it also presents important challenges by adding noise into the user-generated place records for many downstream tasks such as Point of Interest (POI) recommendation, place search, data filtering, and automatic place name database creation that can perform better with high quality data. In [36], authors observed that about 30% of created venues in Whrrl and Foursquare are lacking any meaningful textual descriptions. Based on our observation of data collected from Twitter, many of these place records are personal places (e.g. a user's private home) or entities without any physical location (e.g. online stores).

People in architecture, urban planning, philosophy, and geography have defined and categorized places mainly into four categories [22]:

- ***public places****, places that do not systematically limit the entry of people. Typical examples include public squares, parks, and beaches.*
- **semi-public places** such as restaurants, stores, and other commercial places where entry is not limited as long as one is engaging in the sanctioned activities such as eating, drinking, and shopping.

HT'17, July 4-7, 2017, Prague, Czech Republic.

DOI: http://dx.doi.org/10.1145/3078714.3078722

[1]https://foursquare.com/
[2]https://www.facebook.com/places/
[3]https://www.yelp.com/
[4]https://brightkite.com/
[5]http://gowalla.com/
[6]For example, Foursquare allows users to create new venues via Foursquare website or mobile applications

- ***private places***, that are not open to all. Typical examples are people's homes, gardens, bedrooms.
- ***virtual places***, which do not have an actual physical location, e.g., online shopping stores, chat rooms.

In this work, we use public to also include semi-public places and tag all places in our dataset as one of the three categories: ***public (+ semi-public), private,*** or ***virtual***. Please note that distinction between ***public places*** *and* **semi-public places** could have considerable relevance to some downstream tasks such as place recommendations. For example, public places are essentially free, visiting the other (semi-pubic) might involve a cost of some kind (admission fee, purchasing of items) that makes it less attractive to an unwitting visitor. For this current study, we merge these two categories into one to simplify the task of place-type detection.

Given the high volume of check-ins and existing businesses on LBSNs, even a low rate of private and virtual place creation results in a large number of private and virtual places. As a result, ***private*** and ***virtual*** places may bring irrelevant and ambiguous information to various downstream tasks, which makes automatic place-type detection an important research problem. Despite its practical importance, place type detection is a particularly challenging task for several reasons:

- ***data diversity***– Check-in records contain diverse types of data including time, location, and text. Therefore, due to the heterogeneous nature LBSNs, methods that effectively take all these data types into consideration for place-type detection must be developed.
- ***sparse information*** – When creating a venue, a user is asked to provide a few attributes of the venue, such as the venue's name, address, location, category, zip code, cross street, and etc. However, in many cases attributes such as address, category, zip code, and country are not provided by the users. Moreover, users personal experiences (tips) associated with most of the check-in records are either contain a few words or just empty. Without enough context and background knowledge, it is difficult even for a human to determine whether a given place is public or private in the physical world.
- ***overwhelming noise***– Almost 30% of the check-in records do not contain any meaningful textual descriptions.
- ***ambiguity*** – place names can be ambiguous. So, only relying on place names would be challenging to differentiate between place types. Fortunately, in our dataset we have user check-in activities at various places and times. Therefore, we propose to explore the user behaviours to extract useful pattern and features from check-in records in order to distinguish place types.

Whereas most prior computational studies have focused on place labels (e.g., restaurants food, shopping, hotel travel, arts entertainment) annotation [6, 16, 36], there has been a lack of work looking at place-type detection in LBSNs. To the best of our knowledge, this is the first attempt to solve the problem of place-type detection in LBSNs. For doing so, fundamental issue is identifying and extracting a number of descriptive features for each place type from the available check-in records. Following the idea of [36] for semantic annotation of places, we explore the set of user behaviours and look for unique features of places recorded in the check-in data for the specific task of place-type classification. We know that human behaviours are not completely random [12] and so can be predicted [17]. For example, people often go for cinema on a Friday/at the weekend in the evening. Moreover, people exhibit patterns in their activities, e.g. various places visited by the same person at the same time may be similar (e.g. having the similar type).

Similar to [36], we compute two kind of features: (i) specific patterns (SP) at individual places; and (ii) latent relatedness (LR) among similar places. Features computed from SP, corresponding to a given place, can be derived from all check-ins at that place. We compute features from LR to determine the relatedness among similar places. Since we have only small number of places manually annotated with their type, we can make good use of LR by deriving descriptive features of a given place from its "related" places. To facilitate the extraction of LR features, we adapted a similar strategy proposed in [36] to build a graph of related places (GRP) by exploiting the regularity of user check-in records to similar places. In particular, we explore different graph representations: (i) visitors-place; and (ii) time-place relationship from the user check-in records. We employ different techniques to these graphs to measure their relatedness. Finally, we calculate the probability of the category tag for each place by leveraging the relatedness of places on the graph and treat them as LR features for supervised learning algorithms.

We then implement a supervised ensemble learning framework defined as a multi-objective optimization problem in order to i) obtain accurate classification results even when training evidences are limited; and ii) identify different solutions thus offering a wide range of possible application scenarios. Indeed, depending on the task at hand, precise classification may be required (e.g. filtering) or high recall may be preferred (e.g. ranking of places for recommendation).

Finally, we examine the usefulness of place-type tagging in the context of place name recommendation. In particular, we present a neural network framework to complete the place recommendation task and compare its performance in various scenarios.

2 RELATED WORK

Previous studies in LBSNs can broadly be divided into two different categories: recommendations and place labeling.

Recommendation in LBSNs is basically divided into four different categories [2] : i) location recommendations, which suggest locations (e.g., POIs) or sequential locations (such as travel routes) to a user; ii) user recommendations, which suggest popular users (like local experts), potential friends (i.e., who share similar interests and preferences), or communities, which a user may wish to join due to shared interests and activities; iii) activity recommendations, which refer to activities that a user may be interested taking; iv) content recommendations, which suggest media as photos, videos, and web contents, to the user. Depending on the working methodology and used data attributes, recommender systems in LBSNs can be divided into: a) content-based recommendation, which uses data from a userfis profile and the features of locations; b) link analysis-based recommendation, which applies link analysis models, e.g., hypertext induced topic search (HITS) and PageRank; and c) collaborative filtering (CF) recommendation, which infers a userfis preferences from historical behavior.

Venue recommendation has been the focus of research in LBSNs. Several recommendation systems have been proposed in the literature including [8, 37, 42]. In [8], authors developed GeoSocialDB–a recommender system for providing three services, namely, location-based news feed, location-based news ranking, and location-based recommendation. In particular GeoSocialDB implemented these services as query operators inside a database engine to optimize the query processing performance. An interesting strategy, namely, user-centered collaborative location and activity filtering (UCLAF) method is proposed in [42], to pull many users' data together and apply collaborative filtering to find like-minded users and like-patterned activities at different locations. Authors in [42] modeled the user-location-activity relations with a tensor representation, and proposed a regularized tensor and matrix decomposition solution which can better address the sparse data problem in mobile information retrieval. In line with [42], [37] analyzed location recommendation services for large-scale LBSNs, by exploiting the social and geographical characteristics of users and locations/places. Precisely, they proposed a variant of friend-based collaborative filtering (FCF) technique, namely Geo-Measured FCF (GM-FCF), based on heuristics derived from observed geospatial characteristics in the Foursquare dataset for location recommendation.

Recently, researchers started to explore the content information on LBSNs for POI recommendation. In [13], authors showed that content information in LBSNs can be useful for POI recommendations. In particular, authors studied three types of content information (namely POI properties, User Interests, and Sentiment Indications) and proposed a unified framework to model them to achieve better performance for POI recommendation.

Different from the above mentioned works, several works exist to study sequential location recommendations based on either users' social media post [20, 35] or users' GPS trajectories [5, 39]. A Large volume of works have also been proposed for other categories of recommendations: user recommendations [10, 28, 38], activity recommendations [43], and content recommendations [25, 30].

Place labeling is the process of attaching semantic labels to venues, such as home, work, and school [16]. Place labeling techniques can be categorized mainly into two types: i) Manual; and ii) automatic. There are several prototypes exist that allows end users to manually label the places they visit, such as Reno [31], Connecto [3], and IMBuddy [15]. Automatic place tagging techniques can be classified mainly into two categories: i) rule based; and ii) machine learning based approach. In [44], authors proposed a system that rely on manually designed classification rules to infer the semantic category of a place. Despite effectiveness, this kind of methods require substantial efforts in rule design.

One of the very first attempts to propose a machine learning model that deals with place labeling task is proposed by [21]. The authors developed a system that uses hierarchically structured conditional random fields to generate a model of a person's activities and places. The computational models are learned over features from the locations of nearby restaurants, grocery stores and bus stops as well as the timing of visits. In [6], authors proposed a Hidden Markov Model (HMM)-based Location Extraction algorithm called HLE, which adopts a supervised learning based method for extracting user's daily significant semantic locations using mobile phone data.

Recently, the introduction of Nokia Mobile Data Challenge (MDC) [18] has clearly established the importance of place labeling tasks. The MDC provided labeled data and cell phone logs for 114 people (80 for training, 34 for testing) with an average of 282 days of observation for each one. All of the participants for the place labeling task adopted machine learning techniques and used phone features, including the time and duration of visits, to infer place label.

One of the most influencing work in this direction is proposed by [36], who considered the problem as multi-label classification problem and used supervised classification strategy to tackle the problem. In order to learn the classifier, two groups of features are computed from the check-in records. First group of features are derived from the patterns observed in places with same tag. The second group of feature is computed by exploiting similarities among similar places. These feature sets are used as inputs for the place labeling phase to learn a binary SVM for each tag. Finally, output of all SVM classifiers are assembled to derive the final labels. They conducted a experimental study based on a dataset collected from Whrrl for a period of one month consisting of 5,892 users, 53,432 places and 199 types of tags. Based on Yelp tag hierarchy, they merge those 199 semantic tags into 21 categories to simplify the task of place label annotation. Although these works are valuable in the context of LBSNs, its scope differs from our specific goal of place-type detection.

3 PROBLEM FORMULATION

Let $P = \left\{p_1, p_2, \ldots.., p_{|P|}\right\}$ be the set of places in our dataset, where $|P|$ denotes the total number of places. Each place $p_i \in P$ can be represented as $p_i = \langle \text{name}_i,\ lat_i,\ lon_i,\ A_i \rangle$ that indicates its given name, latitude, longitude, and attributes such as address, location, zip, cross street, and country. Moreover, some additional information is also available in the form of total number of check-ins, time of check-ins etc.. Given all the information for each place p_i, our goal is to predict its place type $t \in \{public, private, virtual\}$.

3.1 Approach Overview

In this section we present an overview of the approach adopted for place-type detection problem. The first step of the algorithm takes care of feature extraction, while the second step deals with place type assignment. While we explore SP in the check-in records of individual place to extract first group of features, the LR between similar places is used to compute descriptive features of a given place compared to its similar places. Supervised learning strategy is used to learn several ternary (public, private, and virtual) classifiers over the two groups of features derived from SP and LR on a set of manually labeled data in the place-tagging phase. Finally, individual decisions of classifiers are combined using a multi-objective ensemble learning framework to achieve higher accuracy and offer robust solutions to the task at hand.

3.2 Features derived from SP

Our motivation is to extract discriminative features from places of similar type. One can expect that at different places, users conduct themselves in accordance with the accepted activities offered by these places. As a consequence, distinct patterns form in the aggregated behaviors of users at different place types. These patterns are embedded in the check-in activities of users in LBSNs.

- *Total Number of Check-ins*: We observed (shown in Figure 1(a)) from the collected dataset that public places (same as restaurants and universities) attract higher numbers of check-ins than private places (e.g. home, private luxury vehicles). Therefore, the number of check-ins is considered as an important feature for the classification of place type.
- *Total number of distinct users*: This feature aims to capture the total number of distinct users who checked-in or were tagged at a specific place.
- *Check-in time in a week*: We examine (as shown in Figure 1(b)) the check-in patterns for different categories of places over the day of a week. We find that users check-in at a university more often on weekdays than at weekends. In contrast, they checked-in to online shopping stores at weekends more frequently than during weekdays.
- *User check-in locations*: We find from our dataset that location distribution patterns of users checking-in at public places is different from those observed for virtual or private places; public places has often attract high volume of check-ins from various locations that are either near or far within same city or region from the place's physical address, while virtual places have check-ins scattered across much wider geographical areas. Therefore, we compute the minimum, maximum, as well as average distance of check-in users at a specific place and consider these values as discriminative features for the classification of places such as online chat rooms and restaurants. To measure the distance between longitude/latitude points, we use the Haversine formula [33] to calculate the great-circle distance between two points, i.e. the shortest distance over the earth's surface.
- *n-grams*: We use 1-3 token sequences. Features are encoded simply as binary indicators regarding whether the n-gram appears in the place names.
- *Place profile*: We observed that places with more complete profile are more likely to be public. We consider, two attributes, namely, 'contact", "cross street". Features are encoded simply as binary indicators regarding whether the entries are there or not.

3.3 Features derived from LR

The rationale behind extraction of features from LR is that people's activities are not completely irregular. For example, we usually go to places for food at lunch/dinner time, visit places for shopping in the late afternoon, and usually return to our home in the evening. Such patterns appear for certain users in our dataset and so we explored these to tag similar places. To record the relatedness among places and compute discriminative features, similar to [36] we built a graph of related places (GRP), where places are linked based on their relatedness, as measured from the information embedded in the user check-ins using the Random Walk and Restart method [32] (RWR). On the GRP, we compute the label probability of each place leveraging the relatedness of places. The derived label probability

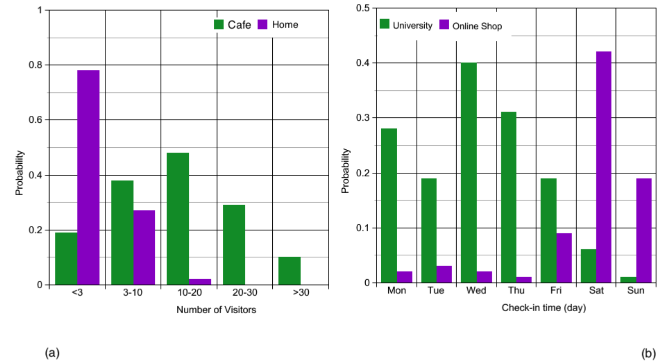

Figure 1: Check-in details at different place types: (a) Number of visitors at Cafe and Home, (b) Distribution of check-in time at University and Online Shop.

is used as a feature for classification. The details of our feature extraction from LR model are as follows.

Graph of Related Places: To facilitate the extraction of features from latent relatedness among similar places, following the idea of [36] we build two graphs: visitor-place and time-place graph. The underlying idea behind visitor-place is that the majority of users more often visit similar places. The motivation behind time-place graph is that the timing of check-ins at similar places may be similar. These graphs can be formally defined as:

- A **visitor-place Graph**, $G_u(V_u, E_u)$, is an undirected bipartite graph. Here, $V_u = U \cup P$, where U and P are the sets of all users and places, respectively, and $E_u = \{e_{i,j} | c(u_i, p_j, .) \in C\}$, where C is the collection of all check-in records and $c(u_i, p_j, .)$ denotes that user u_i has visited place p_i at some time. Each edge $e_{i,j} \in E_u$ is associated with a weight $w_{i,j}$, denoting how often user u_i has visited place p_i. Formally, $w_{i,j} = |\{c(u_i, p_j, h_s)\}|$, where h_s is the time stamp.
- A **time-place Graph**, $G_t(V_t, E_t)$, is an undirected bipartite graph. Here, $V_t = H \cup P$, where H and P are sets of all times (i.e. hours), and places, respectively, and $E_t = \{e_{j,s} | c(., p_j, h_s) \in C\}$, where C is the collection of all check-in records and $c(., p_j, h_s)$ denotes that a user has visited place p_j at time h_s. In this graph, each edge $e_{j,s} \in E_t$ is associated with a weight $w_{j,s}$, denoting how often p_j has been checked in at time h_s. Formally, $w_{j,s} = |\{c(u_i, p_j, h_s)\}|$.

Places are connected indirectly through visitors and times in the graphs described above. To construct the GRP, the relatedness of places from the *visitor-place* and *time-place* graphs needs to be derived. In this experiment, we calculate two relatedness values $r^u_{x,y}$ and $r^t_{x,y}$ for every pair of places p_x, p_y using RWR over the *visitor-place* and *time-place* graphs, respectively, and then merge them into one relatedness value between place nodes in the GRP. Below we only present how our RWR technique is applied on the *visitor-place* graph since the operation in *time-place* graph is similar.

Given a node x, RWR is carried out by randomly following one of its links to another node y in the *visitor-place* graph depending on the transition probabilities of these links, as well as on a probability

a to restart at node x. Our random walk transition matrix consists of two zero matrices, i.e. visitor-visitor matrix (VV) and place-place matrix (PP), and a visitor-place (UP) matrix and transpose UP^T, where the probability of transiting between a place p_j and a visitor u_i is proportional to $w_{i,j}$. The stationary (or steady-state) probabilities of each pair of nodes can be acquired by recursively processing RWR until convergence. The converged probabilities (i.e., relatedness values) give us the long-term visiting rates from any given node to any other node. In this way, we can calculate the relatedness of all pairs of location nodes, denoted by $r^p_{x,y}$ ($\forall p_x, p_y \in P$). Accordingly, we can derive two relatedness values $r^u_{x,y}$ and $r^t_{x,y}$ from *visitor-place* and *time-place* graph, respectively. Afterwards, we calculate the overall relatedness value for each pair of location as equation 1.

$$r^p_{x,y} = \eta r^u_{x,y} + (i - \eta) r^t_{x,y}, \forall p_x, p_y \in P \tag{1}$$

where η is a smoothing factor in the range 0 to 1. Finally, a graph of related place (GRP) is built where each place is connected to places with top-k relatedness values.

Place type label probability estimation: Our dataset contains millions of check-ins and it is challenging to create a sufficient amount of labeled data to cover various cases of *public, private*, and *virtual* places. Therefore, we build GRP which is able to make use of a large amount of unlabeled data to infer the label of a given place from its related places. In order to estimate the label probability of a place to be labeled, we derive the probability from the place tags of its neighbours recursively [23]. Assume N_i be the set of immediate neighbours with edges connecting place p_i, and y_i be a variable denoting a tag of place p_i. For all possible tags $t \in T$, we adopt a method similar to [36] for deriving the final $Pr(y_i = t|N_i)(t \in T)$ for each place p_i. The label probability of p_i is calculated by taking into account both the weighted average of the label probabilities of places in N_i, and the current label probability of p_i itself as equation (2).

$$Pr^{(n+1)}(y_i = t|N_i) = \beta^{(n+1)} \frac{1}{Z} \sum_{p_j \in N_i} r^p_{j,i} Pr^n(y_j = t|N_j) + (1 - \beta_t^{(n+1)}) Pr^{(n)}(y_i = t|N_i) \tag{2}$$

where $Z = \sum_{p_j \in N_i} r^p_{j,i}$ is a normalization term and $r^p_{j,i}$ is the relatedness between places p_j and p_i, and $Pr^{(n)}(y_i = t|N_i)$ denotes the estimation of $Pr(y_i = t|N_i)$ at round n. We also define $\beta_t^{(n+1)} = \beta_t^{(n)} \alpha$, where $\beta_t^{(0)}(t \in T)$ is a constant between 0 and 1, and α is a decay factor, i.e., $0 < \alpha < 1$.

We have initialized the label probability for each place $p_i \in P$ as follows.

$$Pr^{(0)}(y_i = t|N_i) = \begin{cases} 0.5, & \text{if } p_i \in p_{test} \\ 1, & \text{if } p_i \in P - P_{test} \text{ and } t \in T_i \\ 0, & \text{if } p_i \in P - P_{test} \text{ and } t \notin T_i \end{cases}$$

where p_{test} denotes the set of testing places, i.e., unlabeled data that do not have any place type tag. The label probability of a testing place is initialized as 0.5, while the label probability of a manually tagged place is set to 1 or 0 according to the labels. The label probability estimated for a place p_i is treated as the LR feature for supervised learning.

4 LEARNING FRAMEWORK

An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way (typically by weighted or binary voting) to classify new examples [11]. In particular, ensemble learning is known to obtain highly accurate classifiers by combining less accurate ones thus allowing to overcome the training data size problem. There are methods for constructing ensembles in the literature [11]. In this experiment, we propose ensemble learning as a multi-objective optimization (MOO) problem. Our motivations are two-fold. First, [27] showed that MOO strategies demonstrate improved results when compared to single objective solutions and state-of-the-art baselines. Second, MOO techniques propose a set of solutions rather than a single one. As place type tagging can be thought of as an intermediate module in some larger application (e.g. POI recommendation, place search, or database creation), offering different solutions can be a great value.

4.1 MOO Problem Defintion

A definition of multi-objective optimization can be stated as follows: find the vector $\overline{x} = [x_1, x_2, \ldots, x_n]^T$ of decision variables that optimizes O objective functions $\{O_1(\overline{x}), O_2(\overline{x}), \ldots, O_O(\overline{x})\}$ simultaneously which also satisfy user-defined constraints, if any. The concept of domination is also an important aspect of MOO. In case of maximization, a solution $\overline{x_i}$ is said to dominate $\overline{x_j}$ if both conditions (3) and (4) are satisfied.

$$\forall k \in 1, 2, \ldots, O, \quad O_k(\overline{x_i}) \geq O_k(\overline{x_j}) \tag{3}$$

$$\exists k \in 1, 2, \ldots, O, \quad O_k(\overline{x_i}) > O_k(\overline{x_j}) \tag{4}$$

Finally, the set of non-dominated solutions of the whole search space S is called the Pareto optimal front, from which a single solution may be selected based on any suitable criterion.

Ensemble learning can be seen as a vote-based problem. Suppose that one has a total number of N classifiers $\{C_1, C_2, ..., C_N\}$ trained for an M class problem. Then, the vote-based classifier ensemble problem can be defined as finding the combination of votes V per classifier C_i, which will optimize a quality function $F(V)$. V can either represent a binary matrix (binary vote-based ensemble) or a matrix containing real values (real/weighted vote-based ensemble) of size $N \times M$. In case of binary voting, $V(i, j)$ represents whether C_i is permitted to vote for class M_j. $V(i, j) = 1$ is interpreted as the i^{th} classifier being permitted to vote for the j^{th} class, else $V(i, j) = 0$ is interpreted as the i^{th} classifier is not permitted to vote for the j^{th} class. In case of real voting, $V(i, j) \in [0, 1]$ quantifies the weight of the vote of C_i for the class M_j. If a particular classifier is confident in determining a particular class, then more weight should be assigned to that particular pair, otherwise less weight should be attributed. In terms of MOO formulation, the classifier ensemble problem at hand is defined as determining the appropriate combination of votes V per classifier such that objectives $O_1(V)$ and $O_2(V)$ are simultaneously optimized where O_1 = recall and O_2 = precision.

4.2 Evolutionary Procedure

The multi-objective methods used here are based on the search capabilities of the non-dominated sorting genetic algorithm [9].

String Representation: In order to encode the classifier ensemble selection problem in terms of genetic algorithms, we propose to study three different representations.
(1) Simple Classifier Ensemble (SCE): each individual classifier is allowed to vote (or not). The chromosome is of length N and each position takes either 1 or 0 as value.
(2) Binary Vote-based Classifier Ensemble (BVCE): each individual classifier is allowed to vote (or not) for a specific class M_j. The chromosome is of length $N \times M$ and each position takes either 1 or 0 as value.
(3) Real/weighted Vote based Classifier Ensemble (RVCE): all classifiers are allowed to vote for a specific class M_j with a different weight for each class. The chromosome is of length $N \times M$ and each position takes a real value.

Fitness: Each individual chromosome corresponds to a possible ensemble solution V, which must be evaluated in terms of fitness. Let the number of available classifiers be N and their respective individual F-measure values by class F_{ij}, $i = 1 \dots N, j = 1 \dots M$ (i.e. F_{ij} is the F-measure of C_i for class M_j). For a given place p, receiving class M_j is weighted as in equation (5) where the output class assigned by C_i to p is given by $op(p, C_i)$. Note that in the case of SCE, $V(i, j)$ is redefined as $V(i, .)$ and F_{ij} as $F_{i.}$.

$$f(p, M_j) = \sum_{i=1:N \& op(p,C_i)=M_j} V(i, j) \times F_{ij}. \quad (5)$$

Finally, the type of place p is given by $argmax_{M_j} f(p, M_j)$. As such, classifying all places from a development set gives rise to two fitness (or objective) values, which are, respectively, recall (O_1) and precision (O_2) and must be optimized simultaneously.

Optimization and Selection: The multi-objective optimization problem is solved by using the Non-dominated Sorting Genetic Algorithm (NSGA-II) [9]. The most important component of NSGA-II is its elitism operation, where the non-dominated solutions present in the parent and child populations are moved to the next generation. The chromosomes present in the final population provide the set of different solutions to the ensemble problem and represent the Pareto optimal front.
It is important to note that all the solutions are important, representing a different way of ensembling the set of classifiers. However, for the purpose of comparison with other methods, a single solution is required to be selected. For that purpose, we choose the solution that maximizes the F-measure based on its optimized sub-parts recall and precision as shown in equation (6):

$$F\text{-measure} = \frac{2 \times recall \times precision}{recall + precision}. \quad (6)$$

5 CHECK-IN DATA COLLECTION AND LABELING

Since personal check-in information on location-sharing services like Foursquare, Gowalla, and Facebook Places is typically restricted to a user's immediate social circle (and hence unavailable for sampling), we take an approach similar to [7] to collect check-ins. In particular, we sampled location sharing (geo-tagged) Foursquare tagged tweets from Twitter Public Stream[7] for a month (from 1 March 2014 to 1 April 2014) and filtered out non-English tweets. Using this approach, we collected a dataset consisting of 155,27 users who performed 2,737,442 check-ins at 314,650 venues globally. In order to identify the language of a check-in message, we leverage the language detection library developed by Cybozu Lab [29].

Since no annotated place dataset exists, we designed our own annotation task using the crowdsourcing service of CrowdFlower platform.[8] We randomly sampled 10,000 venues and uploaded them to CrowdFlower. In particular, we represent each venue with details such as total number of check-ins, total number of unique users and their locations, tweet text, and asked crowdFlower annotators to decide whether the place is a *public, private*, or *virtual* place. There was a fourth option available to the annotators namely *"Unsure"*, when they are not confident about their decision. Each annotator was presented with detailed annotation instructions. Each venue was annotated by at least 4 annotators. Venues receiving a majority vote (at least 3 or more) for a particular class are considered as gold-standard, with the reminder rejected. The gold-standard data set contains 9,218 venues: *public*=6591; *private*=862; *virtual*=1765.

We follow standard rules of thumb for splitting a sample into a training set, a development set, and a test set. In particular, we divide instances of each place category from the gold-standard into the ratio of 3:1:1 for training, development, and testing, respectively. The final distributions are presented in Table 2.

6 EXPERIMENTS

Experiments for learning are run in a two-step process. First, N = 10 individual classifiers are learned over the features extracted from SP and LR on the training instances. For each classifier C_i, $F_{i.}$ (global F-measure) and F_{ij} (F-measure for class M_j) values are stored. All experiments were run over the Weka platform.[9] Following Weka's denomination, the list of the 10 classifiers is as follows: NaiveBayes, NBTree, MultilayerPerceptron, RandomForest, J48, LMT, RBFNetwork, Logistic, SimpleLogistics, and SMO. In order to assess the quality of each individual classifier, each one was tested on the test set containing 1844 venues. The results of the top-5 classifiers are given in Table 3.

The second step of the experiment is the optimization procedure. For that purpose, we used the development set consisting of 1844 venues. Based on the development set, the evolutionary optimization using NGSA-II is run for three representations (SCE, BVCE, RVCE) and the best solution is selected based on maximum F-measure as defined in equation (6). Performance results are presented in Table 1 and compared to two baseline ensemble techniques (BSL1, BSL2). BSL1 corresponds to Boosting with the single Logistic classifier, and BSL2 is a SVM solution with 10 features, each one corresponding to the output class (i.e., public, private, virtual) of each of the 10 classifiers.

As expected, our methodology significantly outperforms BSL1 and BSL2 in terms of F-measure for the RVCE representation. In particular, BSL1 suffers from the use of a single classifier family

[7] https://dev.twitter.com/docs/streaming-apis/streams/public
[8] http://www.crowdflower.com/
[9] http://www.cs.waikato.ac.nz/ml/weka/

Method	RVCE	BVCE	SCE	BSL1	BSL2
Public (p, r, f1)	(0.77,0.75,0.76)	(0.73,0.71,0.72)	(0.75,0.74,0.74)	(0.67,0.65,0.66)	(0.66,0.65,0.65)
Private (p, r, f1)	(0.79,0.77,0.78)	(0.74,0.71,0.72)	(0.78,0.75,0.76)	(0.68,0.66,0.67)	(0.67,0.67,0.67)
Virtual (p, r, f1)	(0.76,0.74,0.75)	(0.72,0.71,0.71)	(0.75,0.73,0.74)	(0.66,0.64,0.65)	(0.65,0.63,0.64)
Overall (p, r, f1)	**(0.76,0.74, 0.75)**	(0.73,0.71,0.72)	(0.73,0.73,0.73)	(0.68,0.65,0.66)	(0.66,0.65,0.65)

Table 1: Precision (p), recall (r), and f-measure (f1) achieved by different ensemble learning strategies for the place-type tagging task.

Dataset	Public	Private	Virtual	Total
Training	3955	516	1059	5530
Development	1318	173	353	1844
Test	1318	173	353	1844

Table 2: Distribution of places in training, development, and test sets.

Classifiers	Precision	Recall	F-measure
Logistic	**0.65**	**0.64**	**0.64**
SMO	0.63	0.64	0.64
RandomForest	0.59	0.58	0.58
LMT	0.58	0.56	0.57
SimpleLogistics	0.56	0.55	0.56

Table 3: Results of single learning strategies.

while BSL2 cannot generalize over the small amount of training data. Moreover, the most fine-tuned strategy in terms of ensemble learning demonstrates improved results when compared to coarse-grain solutions. Improvements of 3% and 2% are shown against BVCE and SCE, respectively. In Table 4, we provide some examples of venues tagged as *public*, *private*, and *virtual* by the RCVE representation.

Public	Private	Virtual
CST Brands Corner Store	My Grand Villa	BlogtoRead
Central Park West, NYC	Kia Optima Ex.	MoneyGram
Astoria Plaza, HK	Static Caravan	TransferWise
CDG Airport	My Farm house	Boohoo
Wells Fargo	Apt. Home	Lavish Alice

Table 4: Examples of automatically tagged *public*, *private*, and *virtual* places.

In order to understand the spectrum of the different solutions on the Pareto front, we present in Table 5 three different situations: the solution that maximizes precision (line 1), the solution that maximizes recall (line 2) and the solution that maximizes F-measure (line 3). Results show that high overall performances are provided by every solution. However, depending on the application at hand, one may expect to find a better tuned configuration.

Measures	Recall	Precision	F-measure
Max precision	0.69	**0.79**	0.73
Max recall	**0.78**	0.70	0.73
Max F-measure	0.74	0.76	**0.75**

Table 5: Precision and recall spectrum.

7 APPLICATION

We propose to test the usefulness of place-type tagging in the context of POI recommendation in LBSNs since it can be beneficial for many scenarios. including helping users explore attractive locations, as well as helping LBSNs to increase revenues by providing users with intelligent location services and location-aware advertisements. Note that our ultimate goal here is to examine whether place-tagging can improve the overall performance of a POI recommendation system.

With the available check-in records, existing recommendation approaches can be employed for POI recommendation in LBSNs by treating POIs as items. These approaches are mainly centred around collaborative filtering and matrix/tensor factorization [40]. In this experiment, we used neural networks to complete the task for several reasons: (i) it is natural to consider the POI recommendation problem as a sequential prediction problem since a user's visiting history can be considered as a sequence of venues, (ii) neural networks have been successfully applied to tackle sequence prediction problems [24, 41], and (iii) neural networks can learn richer representations compared to matrix factorization, and are more powerful in modeling complex relationships [4].

We formulate the POI recommendation problem as a sequential prediction problem [26]. Let U=$\{u_1, u_2, ..u_{|U|}\}$ be the user set and P=$\{p_1, p_2, ..p_{|P|}\}$ be the set of venues/places in our data set. For each user u, there is a sequence of places visited by the user represented as S_u=$(B_u^1, B_u^2, ..B_u^{t-1})$, where B_u^t is a set of venues/places visited by user at time t (we considered time t of granularity one day). The sequential prediction problem is to predict B_u^t for each user u, given S_u.

In this work, we propose to follow the work of [34] and introduce neural network-based recommender (NNR) framework consists of three layers: embedding, hidden, and output layers. The embedding layer takes a user id and the venues in the user's last k baskets. In the dataset, following the notion of item recommendation, *basket* is defined as a list of places visited by a user on one day. First, the inputs are transformed into a distributed representation where each user and place are represented as a vector $u \in \mathbb{R}^{d_u}$ and $v \in \mathbb{R}^{d_p}$, respectively. We obtain the user matrix $\mathbb{U} \in \mathbb{R}^{d_u * |U|}$ and place matrix $\mathbb{V} \in \mathbb{R}^{d_p * |P|}$ by putting all user and place vector together. Both $\mathbb{U}$ and $\mathbb{V}$ are learned during training. The output of the embedding layer is the concatenation of the user's and the place's representation and can be represented as $h_1 = \mathbb{R}^{d_u + k * d_p}$. It can be considered as representative of both the user's personal interest (what places the user likes) and sequential relatedness between places (the effect of places visited before compared to places visited next).

The next layer in the proposed neural network model is a non-linear hidden layer, which transforms h_1 to a hidden representation h_2 with dimensions l. Here $h_2 = tanh(W_1 h_1 + b_1)$, where $W_1 \in \mathbb{R}^{l * |h_1|}, b_1 = \mathbb{R}^{l * 1}$ are parameters to be learned. $tanh$, the most commonly used activation function in neural networks, is considered for this experiment. Finally, the output layer is a *softmax* layer, which produces the probabilities of the next places:

$$s = W_2 h_2 + b_2, \Pr(i_j \in B_t | u, B_{t-1}, .., B_{t-k}) = \frac{e^{s_{ij}}}{\sum_{n=1}^{|P|} e^{s_{in}}}$$

where $W_2 \in \mathbb{R}^{|P| * l}, b_2 \in \mathbb{R}^{|p| * 1}$ are parameters to be learned.

Our model has several advantages over existing state-of-the-art methods such as collaborative filtering and matrix/tensor factorization. Firstly, it can successfully model longer dependencies by varying the window size k (i.e. by taking a list of places visited by a user over a longer period of time) of the embedding layer, while other methods only capture the influence of recently visited places. Secondly, the embedding layer is flexible and capable of handling other features such as user and place attributes other from user ids and place ids. Finally, the hidden layer gives the freedom to model more complex relationships between users and places.

We conduct experiments to access the effectiveness of our approach on the check-in records collected for the period 1 March 2014 to 1 April 2014. Firstly, we split the dataset D into two non-overlapping sets: a training set D_{train} and a test set D_{test}. Again we followed standard procedure for splitting a sample into a training set, a development set, and a test set. In particular, the splitting is done by putting places visited by each user in the last week of our collection period into D_{test}, and the remaining ones into D_{train}. We used training data to create recommendations, and then we checked whether a user had followed the recommendation during the testing period, i.e. a fixed time period of one week. For each user we discarded all places from the test set (and corresponding predictions) that this user had already visited in the past, under the assumption that recommending to users new locations that they have never been to before is of greater importance recommending some already visited location. Note that this makes the prediction task much harder, as simply recommending already visited places is trivial.

Based on the test set, the dimensions of $h_2(i.e., l)$, user vector (i.e.,d_u), and place vector (i.e.,d_p) are optimized. We recommend the top-C places for each user, denoted as $\hat{B_u^t}$, and use recall and precision over all test baskets using the top-5, -10 and -20 lists for evaluation. Precision (p) and recall (r) are defined as follows.

$$p = \frac{\sum_u |B_u^t \cap \hat{B_u^t}|}{|\mathbb{U}| * C}, r = \frac{\sum_u |B_u^t \cap \hat{B_u^t}|}{\sum_u |B_u^t|}$$

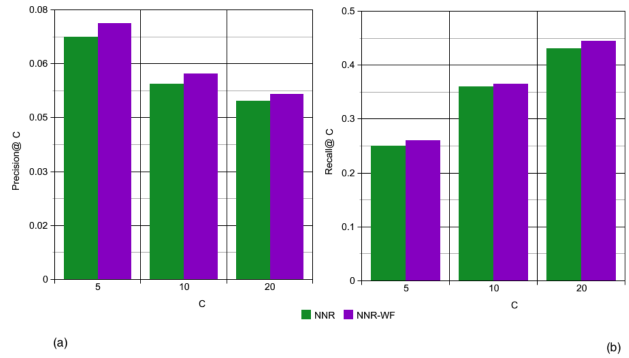

Figure 2: Precision and recall of neural networks based recommender with and without *private* place filtering. NNR represent model without *private* place filtering while the model, denoted as NNR-WF, includes *private* place filtering step.

In order to check the usefulness of place-type tagging in POI recommendation, an additional filtering step is introduced. Specifically, we removed places tagged as *private* from the dataset and measured the performance of NNR. Note that for filtering of *private* places, high precision is preferred over recall and so the solution presented in the first line of Table 5 is considered. Comparative precision and recall scores are presented in Figure 2(a) and (b), respectively. The figures show that the neural network recommender with the *private* place filtering step included (denoted as NNR-WF in the figures) is slightly better in terms of precision and recall than its counterpart. Note that only unvisited POIs are recommended for each user which explains the somewhat low performance of all methods. Results also indicate that place-type tagging can be utilized in the data preprocessing step to enhance the performance of POI recommendation.

8 CONCLUSIONS

To the best of our knowledge, we presented the first work on detection of place-type in location-based social networks. We adopted supervised machine learning strategy to tackled the problem of tagging places as *public, private,* or *virtual.* Due to the small amount of 'gold standard' training data, we proposed an ensemble learning solution, the underlying idea of which is to reduce bias by combining multiple classifiers instead of relying on a single one. In particular, recently developed multi-objective-based ensemble techniques have been applied to improve overall accuracy. By extracting effective features from check-in records and exploring large amounts of unlabeled data, our work achieves reasonable accuracies for all place types. Finally, we proposed to take a look at how recommender systems can benefit from this task. Precisely, we examined neural network-based POI recommendation and reported comparative results where place-type tagging is considered as an intermediate module. In future, we would like to consider other neural networks which can model longer sequential dependencies and use additional features such as user's immediate social circle and place attributes for POI recommendation.

ACKNOWLEDGMENTS

The ADAPT Centre for Digital Content Technology is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

REFERENCES

[1] Jie Bao, Yu Zheng, David Wilkie, and Mohamed Mokbel. 2015. Recommendations in location-based social networks: a survey. *GeoInformatica* 19, 3 (2015), 525–565. DOI: https://doi.org/10.1007/s10707-014-0220-8
[2] Jie Bao, Yu Zheng, David Wilkie, and Mohamed Mokbel. 2015. Recommendations in location-based social networks: a survey. *GeoInformatica* 19, 3 (2015), 525–565.
[3] Louise Barkhuus, Barry Brown, Marek Bell, Scott Sherwood, Malcolm Hall, and Matthew Chalmers. 2008. From Awareness to Repartee: Sharing Location Within Social Groups. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 497–506. DOI: https://doi.org/10.1145/1357054.1357134
[4] Yoshua Bengio. 2009. Learning deep architectures for AI. *Foundations and trends® in Machine Learning* 2, 1 (2009), 1–127.
[5] Kai-Ping Chang, Ling-Yin Wei, Mi-Yeh Yeh, and Wen-Chih Peng. 2011. Discovering Personalized Routes from Trajectories. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks (LBSN '11)*. ACM, New York, NY, USA, 33–40. DOI: https://doi.org/10.1145/2063212.2063218
[6] Z. Chen, Y. Chen, S. Wang, and Z. Zhao. 2012. A supervised learning based semantic location extraction method using mobile phone data. In *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, Vol. 3. 548–551. DOI: https://doi.org/10.1109/CSAE.2012.6273012
[7] Zhiyuan Cheng, James Caverlee, Kyumin Lee, and Daniel Z Sui. 2011. Exploring Millions of Footprints in Location Sharing Services. *ICWSM* 2011 (2011), 81–88.
[8] Chi-Yin Chow, Jie Bao, and Mohamed F. Mokbel. 2010. Towards Location-based Social Networking Services. In *Proceedings of the 2Nd ACM SIGSPATIAL International Workshop on Location Based Social Networks (LBSN '10)*. ACM, New York, NY, USA, 31–38. DOI: https://doi.org/10.1145/1867699.1867706
[9] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 181–197.
[10] Peter DeScioli, Robert Kurzban, Elizabeth N Koch, and David Liben-Nowell. 2011. Best friends: Alliances, friend ranking, and the MySpace social network. *Perspectives on Psychological Science* 6, 1 (2011), 6–8.
[11] Thomas G Dietterich. 2000. Ensemble Methods in Machine Learning. In *Multiple classifier systems*. Springer, 1–15.
[12] Nathan Eagle and Alex Sandy Pentland. 2009. Eigenbehaviors: identifying structure in routine. *Behavioral Ecology and Sociobiology* 63, 7 (2009), 1057–1066. DOI: https://doi.org/10.1007/s00265-009-0739-0
[13] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2015. Content-aware Point of Interest Recommendation on Location-based Social Networks. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15)*. AAAI Press, 1721–1727. http://dl.acm.org/citation.cfm?id=2886521.2886559
[14] Nevena Golubovic, Chandra Krintz, Rich Wolski, Sara Lafia, Thomas Hervey, and Werner Kuhn. 2016. Extracting Spatial Information from Social Media in Support of Agricultural Management Decisions. In *Proceedings of the 10th Workshop on Geographic Information Retrieval (GIR '16)*. ACM, New York, NY, USA, Article 4, 2 pages. DOI: https://doi.org/10.1145/3003464.3003468
[15] Gary Hsieh, Karen P. Tang, Wai Yong Low, and Jason I. Hong. 2007. *Field Deployment of IMBuddy: A Study of Privacy Control and Feedback Mechanisms for Contextual IM*. Springer Berlin Heidelberg, Berlin, Heidelberg, 91–108. DOI: https://doi.org/10.1007/978-3-540-74853-3_6
[16] John Krumm, Dany Rouhana, and Ming-Wei Chang. 2015. Placer++: Semantic place labels beyond the visit. In *Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on*. IEEE, 11–19.
[17] Nicholas D Lane, Ye Xu, Hong Lu, Andrew T Campbell, Tanzeem Choudhury, and Shane B Eisenman. 2011. Exploiting social networks for large-scale human behavior modeling. *IEEE Pervasive Computing* 10, 4 (2011), 45–53.
[18] Juha K Laurila, Daniel Gatica-Perez, Imad Aad, Olivier Bornet, Trinh-Minh-Tri Do, Olivier Dousse, Julien Eberle, Markus Miettinen, and others. 2012. The mobile data challenge: Big data for mobile computing research. In *Pervasive Computing*.
[19] Yanhua Li, Moritz Steiner, Limin Wang, Zhi-Li Zhang, and Jie Bao. 2013. Exploring venue popularity in foursquare. In *INFOCOM, 2013 Proceedings IEEE*. IEEE, 3357–3362.
[20] Defu Lian and Xing Xie. 2011. Learning Location Naming from User Check-in Histories. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '11)*. ACM, New York, NY, USA, 112–121. DOI: https://doi.org/10.1145/2093973.2093990
[21] Lin Liao, Dieter Fox, and Henry Kautz. 2007. Extracting Places and Activities from GPS Traces Using Hierarchical Conditional Random Fields. *Int. J. Rob. Res.* 26, 1 (Jan. 2007), 119–134. DOI: https://doi.org/10.1177/0278364907073775
[22] David Lyon. 2006. *Theorizing surveillance*. Routledge.
[23] Sofus A Macskassy. 2007. Improving learning in networked data by combining explicit and mined links. *In Conference on Artificial Intelligence (AAAI)* (2007), 590–595.
[24] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model.. In *Interspeech*, Vol. 2. 3.
[25] Mohamed Mokbel, Jie Bao, Ahmed Eldawy, Justin Levandoski, and Mohamed Sarwat. 2011. Personalization, socialization, and recommendations in location-based services 2.0.
[26] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. ACM, 811–820.
[27] Sriparna Saha and Asif Ekbal. 2013. Combining Multiple Classifiers Using Vote Based Classifier Ensemble Technique for Named Entity Recognition. *Data Knowledge Engineering* 85 (2013), 15–39.
[28] Salvatore Scellato, Anastasios Noulas, Renaud Lambiotte, and Cecilia Mascolo. 2011. Socio-spatial properties of online location-based social networks. (2011).
[29] Nakatani Shuyo. 2010. Language detection library for java. (2010).
[30] Ana Silva and Bruno Martins. 2011. Tag Recommendation for Georeferenced Photos. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks (LBSN '11)*. ACM, New York, NY, USA, 57–64. DOI: https://doi.org/10.1145/2063212.2063229
[31] Ian Smith, Sunny Consolvo, Anthony Lamarca, Jeffrey Hightower, James Scott, Timothy Sohn, Jeff Hughes, Giovanni Iachello, and Gregory D. Abowd. 2005. *Social Disclosure of Place: From Location Technology to Communication Practices*. Springer Berlin Heidelberg, Berlin, Heidelberg, 134–151. DOI: https://doi.org/10.1007/11428572_9
[32] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. 2008. Random walk with restart: fast solutions and applications. *Knowledge and Information Systems* 14, 3 (2008), 327–346. DOI: https://doi.org/10.1007/s10115-007-0094-2
[33] Chris Veness. 2011. Calculate distance and bearing between two Latitude/Longitude points using Haversine formula in JavaScript. *Movable Type Scripts* (2011).
[34] Shengxian Wan, Yanyan Lan, Pengfei Wang, Jiafeng Guo, Jun Xu, and Xueqi Cheng. 2015. Next Basket Recommendation with Neural Networks.
[35] Ling-Yin Wei, Yu Zheng, and Wen-Chih Peng. 2012. Constructing Popular Routes from Uncertain Trajectories. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*. ACM, New York, NY, USA, 195–203. DOI: https://doi.org/10.1145/2339530.2339562
[36] Mao Ye, Dong Shou, Wang-Chien Lee, Peifeng Yin, and Krzysztof Janowicz. 2011. On the Semantic Annotation of Places in Location-based Social Networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*. ACM, New York, NY, USA, 520–528. DOI: https://doi.org/10.1145/2020408.2020491
[37] Mao Ye, Peifeng Yin, and Wang-Chien Lee. 2010. Location Recommendation for Location-based Social Networks. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '10)*. ACM, New York, NY, USA, 458–461. DOI: https://doi.org/10.1145/1869790.1869861
[38] Josh Jia-Ching Ying, Eric Hsueh-Chan Lu, Wang-Chien Lee, Tz-Chiao Weng, and Vincent S. Tseng. 2010. Mining User Similarity from Semantic Trajectories. In *Proceedings of the 2Nd ACM SIGSPATIAL International Workshop on Location Based Social Networks (LBSN '10)*. ACM, New York, NY, USA, 19–26. DOI: https://doi.org/10.1145/1867699.1867703
[39] Hyoseok Yoon, Yu Zheng, Xing Xie, and Woontack Woo. 2012. Social itinerary recommendation from user-generated digital trails. *Personal and Ubiquitous Computing* 16, 5 (2012), 469–484. DOI: https://doi.org/10.1007/s00779-011-0419-8
[40] Yonghong Yu and Xingguo Chen. 2015. A Survey of Point-of-Interest Recommendation in Location-Based Social Networks. In *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*.
[41] Yuyu Zhang, Hanjun Dai, Chang Xu, Jun Feng, Taifeng Wang, Jiang Bian, Bin Wang, and Tie-Yan Liu. 2014. Sequential Click Prediction for Sponsored Search with Recurrent Neural Networks. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.
[42] Vincent W. Zheng, Bin Cao, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative Filtering Meets Mobile Recommendation: A User-centered Approach. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI'10)*. AAAI Press, 236–241. http://dl.acm.org/citation.cfm?id=2898607.2898645
[43] Vincent W. Zheng, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative Location and Activity Recommendations with GPS History Data. In *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*. ACM, New York, NY, USA, 1029–1038. DOI: https://doi.org/10.1145/1772690.1772795
[44] Yin Zhu, Erheng Zhong, Zhongqi Lu, and Qiang Yang. 2013. Feature Engineering for Semantic Place Prediction. *Pervasive Mob. Comput.* 9, 6 (Dec. 2013), 772–783. DOI: https://doi.org/10.1016/j.pmcj.2013.07.004