# A Random Walk Around the City: New Venue Recommendation in Location-Based Social Networks

Anastasios Noulas, Salvatore Scellato, Neal Lathia, Cecilia Mascolo

Computer Laboratory
University of Cambridge
Email: Firstname.Lastname@cl.cam.ac.uk

*Abstract*—The popularity of location-based social networks available on mobile devices means that large, rich datasets that contain a mixture of *behavioral* (users visiting venues), *social* (links between users), and *spatial* (distances between venues) information are available for mobile location recommendation systems. However, these datasets greatly differ from those used in other online recommender systems, where users explicitly rate items: it remains unclear as to how they capture user preferences as well as how they can be leveraged for accurate recommendation.

This paper seeks to bridge this gap with a three-fold contribution. First, we examine how venue discovery behavior characterizes the large check-in datasets from two different location-based social services, Foursquare and Gowalla: by using large-scale datasets containing both user check-ins and social ties, our analysis reveals that, across 11 cities, between 60% and 80% of users' visits are in venues that were not visited in the previous 30 days. We then show that, by making constraining assumptions about user mobility, state-of-the-art filtering algorithms, including latent space models, do not produce high quality recommendations. Finally, we propose a new model based on personalized random walks over a user-place graph that, by seamlessly combining social network and venue visit frequency data, obtains between 5 and 18% improvement over other models. Our results pave the way to a new approach for place recommendation in location-based social systems.

## I. INTRODUCTION

The increasing adoption of location-enabled smartphones has given rise to a number of social services (such as Foursquare and Google Places) that people can use to trace, annotate, and share their experiences about the locations they visit as they navigate through their daily lives. Users notify their friends of the place where they are with a *check-in*, leaving a digital trace of their movements. These services therefore now hold and collect huge datasets that can link users' mobility to both their social connections and the spatial layout of their cities; providers are also beginning to apply a host of machine learning approaches to their data in order to turn check-in services into fully fledged location recommendation systems.

The most prominent problem that challenges building a recommender system in this setting is that the relationship between check-in, social, and spatial data—in terms of understanding how these properties relate to people discovering new places to visit—remains unclear. This has two implications: first, while recommender systems have proven to excel in web settings [1], they have historically operated with ordinal rating data where spatial properties tend to not matter and users have the ability to provide negative feedback. Instead, check-in data only counts users' visits to venues, which are also inherently spread over a geographic space. Second, recommender systems have traditionally operated under the sole assumption of like-mindedness (i.e., historically similar users will continue to have shared preferences). Instead, there are a wide range of reasons why mobile users may want to visit a place (e.g., visiting friends, attending an event, touring culturally significant locations); applying the state-of-the-art in web recommendation to this new context will inexorably exclude a host of features that this data contains.

In this paper, we tackle the problem of building a recommender system for previously unvisited venues from behavioral, social, and spatial data. To do so, we seek to answer the following questions:

1) **How often do people tend to visit new places?** In Section III, we analyze two check-in services' datasets. We discover that between 60-80% of users' check-ins are to venues that have *not* been visited in the previous month; these datasets contain granular representations of irregular behavior beyond daily routines.
2) **What assumptions do web-filtering algorithms make about human mobility?** Section IV describes a host of algorithms—ranging from content-based, social, and collaborative filtering (with neighborhood and latent space models) that have been used to build web recommender systems. We demonstrate that each method has a unique underlying assumption about how people move, which necessarily excludes alternatives when computing recommendations. Furthermore, we show that none of these methods outperform a simple popularity baseline.
3) **How can recommendation quality be improved by combining the different sources of data?** In Section V we propose a generalizable method based on a personalized random walk with restart on a user-place network. It seamlessly and simultaneously combines all the available signals into a high-dimensional graph: such structure takes into account the variety of means through which users are exposed to new venues.

Finally, through an extensive evaluation, we discuss how our approach based on random walks obtains between 5 and 18% improvement over those machine learning algorithms used in web contexts (Section VI). To our knowledge, this is the first study to directly compare and propose how to combine the variety of user preference signals that location-based services collect about their users for new venue recommendation.

## II. RELATED WORK

There are two main research threads related to our work: (1) recommender systems and (2) mobility prediction using spatial-social networks and geolocation data.

There is a wide range of research on the data mining algorithms that form the basis of recommender systems [1]. From a data mining perspective, these systems take as input a set of users' preferences, such as ratings, and aim to predict the preference values for items that have yet to be rated, and fall under the umbrella term *collaborative filtering*. In recent years, work has centered around datasets from the web, namely movies (e.g., Netflix [11]) and music (Yahoo Music[1]); these datasets have characteristic features (such as high sparsity) that pose challenges to the design of accurate preference estimators. Matrix factorization has become a popular approach for collaborative recommender systems [3], due to its robustness in the face of sparse data; moreover, hybrid composites of predictors have recently been awarded for their ability to improve rating prediction [11].

While, historically, users' ratings were considered the sole necessary input data for building recommender systems, there is increasing attention to a variety of other signals that may aid learning algorithms' accuracy. These include temporal features of the data [11] and social network links between users [9]. This broad approach, which relies on augmenting recommendation systems with a more granular picture of the (social, temporal, spatial) setting of the users, has been named *context-aware* recommendation [1]. In the domain that we investigate here, recent work has used both sensors and user-activity data to improve recommendations. For the former, GPS data has been used for location and travel recommendation [25], and mobile phone call records have been used for social event recommendation [18]. The latter group, instead, includes using geo-tagged photos for itinerary recommendation, mining interesting locations, and inferring users' trips [16]. The data that we examine in this paper, which is based on explicit *check-ins* to locations, falls into this latter category (although GPS sensor data may be used to validate the check-in).

The interest in mobility data spans beyond the domain of location recommendation. For example, recent work has used mobile phone and check-in data to infer the structure of friendship networks [7] and the relation between online social networks and physical location [6], a prediction framework which can then be used to recommend social connections to users based on their physical mobility [17], [22]. The interplay between user mobility and their online social connections

has also been analyzed extensively using data extracted from location-based social networks [4], [19]. Prediction algorithms have also been designed to exploit any regularity in users' movements and forecast their future position. A large category of those are based on Markov models [13], while other methodologies include sequence pattern matching [12] and time series analysis [21]. We highlight however, that the *nature* of the problem we are addressing here is different to the mobility prediction task that has been tackled by the above works. Our aim is to recommend new, unvisited venues; models that capture the periodicity of user mobility patterns are not considered in the scope of this paper.

The recent literature, overall, reflects the success of online recommender systems; it is clear that smartphones, as they gain greater traction and popularity, will be the bridge that enables recommender systems to be used in the wild as much as they are used online. However, whether algorithms that have been successful online can be applied to these new domains remains an open question: in the following, by investigating one facet of location recommendation, we show that state-of-the-art collaborative filtering can be outperformed by a hybrid model that aggregates and learns from a range of data about users.

## III. DATA ANALYSIS

In this section, we first describe the publicly available check-in data that we collected for 11 cities across the world (Section III-A). We then analyze the properties of our data sets and we investigate to what extent users visit new places when they use these location-based services. Our main finding is that a large fraction of the visited places are *new* places, which highlights the importance of offering high quality recommendations of new venues in those systems. (Section III-C).

### A. Data Collection

The check-in data we collected spans two different popular location-based services: Foursquare and Gowalla. We restrict our analysis to the 11 most popular cities across both services: this allows us to (a) focus on where these services are most used and (b) restrict our prediction space to areas with the highest venue availability, which maximizes the number of candidate venues that can be recommended.

**Foursquare** was created in 2009 and it has quickly become the most popular location-based service, with more than 15 million users as of January 2012. Per-user Foursquare check-in data is not directly accessible: however, users can opt to share their check-ins publicly on Twitter. We thus were able to crawl for publicly-available check-ins via Twitter's streaming API[2]. Note that we can only access those check-ins that users explicitly chose to share on Twitter, although users have the possibility to set this option as default. The Foursquare dataset contains 35 million check-ins made by 925,000 users across 5 million venues globally and in a period of 5 months (May 27th to November 2nd 2010).

---

| City | $N$ | $M$ | $C$ | $\langle c_u \rangle$ | $\langle c_l \rangle$ |
|---|---|---|---|---|---|
| Austin | 2144 | 3758 | 15665 | 7.3 | 4.2 |
| Boston | 3830 | 2763 | 14730 | 3.8 | 5.3 |
| Dallas | 2418 | 3338 | 13779 | 5.7 | 4.1 |
| Denver | 2097 | 2342 | 10402 | 5.0 | 4.4 |
| London | 7242 | 6609 | 24778 | 3.4 | 3.7 |
| Los Angeles | 8178 | 6918 | 32025 | 3.9 | 4.6 |
| New York | 16131 | 16554 | 93309 | 5.8 | 5.6 |
| Paris | 3091 | 4345 | 13086 | 4.2 | 3.0 |
| San Francisco | 6493 | 6478 | 31070 | 4.8 | 4.8 |
| Seattle | 3493 | 4398 | 20128 | 5.8 | 4.6 |
| Seoul | 9491 | 4284 | 35540 | 3.7 | 8.3 |

TABLE I

AVERAGE PROPERTIES OBSERVED ON FOURSQUARE OVER A PERIOD OF ONE MONTH: TOTAL NUMBER OF USERS ($N$), PLACES ($M$) AND CHECK-INS $C$, AVERAGE NUMBER OF CHECK-INS PER USER ($\langle c_u \rangle$) AND PER PLACE ($\langle c_l \rangle$).

| City | $N$ | $M$ | $C$ | $\langle c_u \rangle$ | $\langle c_l \rangle$ |
|---|---|---|---|---|---|
| Austin | 4008 | 13110 | 76151 | 19.0 | 5.8 |
| Boston | 1050 | 5214 | 15026 | 14.3 | 2.9 |
| Dallas | 3494 | 14586 | 52403 | 15.0 | 3.6 |
| Denver | 1139 | 3188 | 10219 | 9.0 | 3.2 |
| London | 2139 | 13510 | 39696 | 18.6 | 2.9 |
| Los Angeles | 2868 | 12172 | 34659 | 12.1 | 2.8 |
| New York | 2659 | 10903 | 32467 | 12.2 | 3.0 |
| Paris | 480 | 1875 | 4165 | 8.7 | 2.2 |
| San Francisco | 3199 | 13389 | 47128 | 14.7 | 3.5 |
| Seattle | 1595 | 8090 | 26538 | 16.6 | 3.3 |
| Seoul | 336 | 2222 | 3973 | 11.8 | 1.8 |

TABLE II

AVERAGE PROPERTIES OBSERVED ON GOWALLA OVER A PERIOD OF ONE MONTH: TOTAL NUMBER OF USERS ($N$), PLACES ($M$) AND CHECK-INS $C$, AVERAGE NUMBER OF CHECK-INS PER USER ($\langle c_u \rangle$) AND PER PLACE ($\langle c_l \rangle$).

As we did not have access to the actual social network of Foursquare users, we assume that a social link exists between a pair of users if they follow each other on Twitter (thus, they have each explicitly added one another). While this assumption does not guarantee that the social network we create is fully representative of the one in Foursquare, we expect two users which are connected in Twitter are likely to share similar place preferences. In essence, the Foursquare data that we operate with below is a sparse, public sample of the check-ins and social links that Foursquare holds about its users. We estimate, based on Foursquare's public information regarding their user base, that this sample contains between 20% to 25% of the entire Foursquare user base at collection time[3].

**Gowalla** is a location-based social service created in 2009, which has been discontinued since its acquisition by Facebook at the end of 2011. The Gowalla data is a complete snapshot of the service obtained in August 2010, collected via the public API. The entire dataset contains 12,846,151 check-ins made by 216,734 active users, that is, users with at least one check-in made since they joined the service: these check-ins took place across 1,421,262 million venues over about 18 months. It also contains all social links between users, which amounts to 736,778 friendships. Each check-in contains the following fields: the unique user id, the time, its geographic location encoded as latitude and longitude coordinates and the venue's category. In order to assign places from the two venue databases to a specific city we have followed the following methodology. Foursquare venues specify locality information (city, province, street), available through the service's API, thus the assignment was straightforward. In the case of Gowalla, we have assigned a place to a city if it lies within 30 km of its geographic center.[4] This procedure allows us to compare spatially similar sets of places across the two services.



(a) Foursquare Users   (b) Gowalla Users
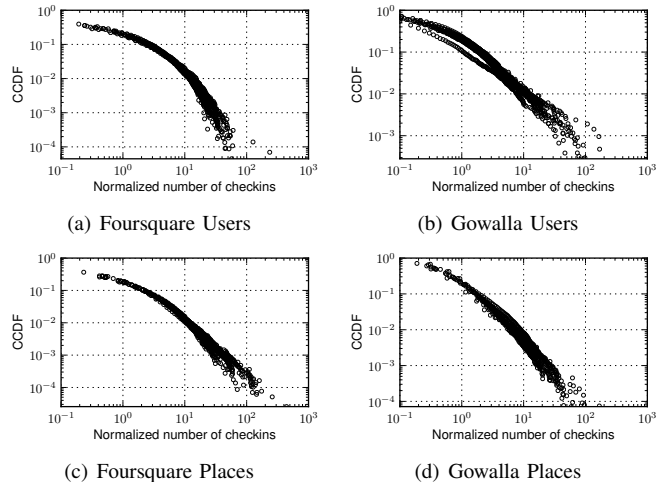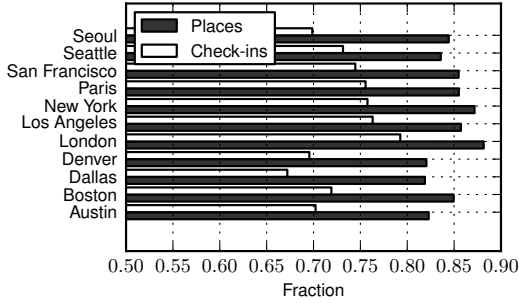
(c) Foursquare Places   (d) Gowalla Places

Fig. 1. Complementary Cumulative Distribution Function of the number of check-ins per user and per-place 1-month of Foursquare (a, c) and Gowalla (b, d) data. Each distribution is normalized with respect to the average value. We consider 1 month of data, but the probability distributions do not change significantly across different snapshots.
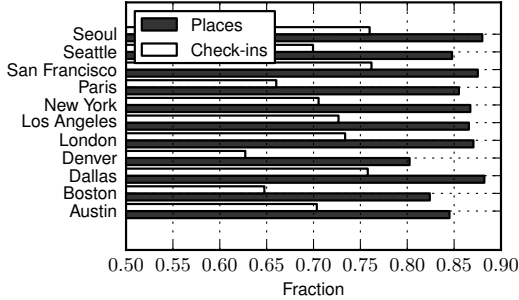
### B. Notation

We now introduce the notation that we will use throughout the paper. We consider a sample of check-in data over a pre-determined temporal period. Each such temporal *snapshot* $t$ contains a set $U$ of $N$ of users and a set $L$ of $M$ places, with each place belonging to a category crowdsourced by the services' users. We represent with $c_{ij}$ the number of check-ins that user $i$ has done at place $j$. The entirety of a user's check-ins in the sample are represented by the vector $\vec{c}_i = (c_{i1}, c_{i2}, \ldots, c_{iM})$. We use $\Phi_j$ to indicate the set of users who have checked-in at place $j$ and $\Theta_i$ for the set of all places where user $i$ has checked-in.

Social ties between users are represented as an undirected graph $G = (V, E)$, with the set of nodes $V = U$ and the set of edges $E$ composed of pairs of users that are present in each other's friend lists in snapshot $t$. We denote with $\Gamma_i$ the set of users connected to user $i$ in graph $G$, with $|\Gamma_i|$ being the number of friends of $i$ in the snapshot.

---

[3]http://mashable.com/2010/08/29/foursquare-3-million-users/

[4]The geographic center has been set according to the median latitude and longitude values across the city's places in the Foursquare dataset.

(a) Foursquare



(b) Gowalla

Fig. 2. Fraction of visited places that were not visited in the last 30 days and fraction of check-ins to such new places, for cities in Foursquare (a) and Gowalla (b).

## C. The Importance of New Places

We now examine a number of properties that the datasets share and study how check-ins to places are distributed. We discover that users tend to visit places they have not visited: between 60% and 80% of check-ins occur at places which were not visited before by an individual user.

Tables I and II present the basic properties of the Foursquare and Gowalla datasets respectively during the same month (August, 2010). The 11 cities differ widely in terms of monthly users: Foursquare has about 22,000 active users each month in New York but only 3,200 in Denver; Gowalla's most popular city, instead, is Austin (where the company was launched) with about 4,000 active users. In general, Gowalla has a smaller number of users and places in each city compared to Foursquare, which reflects the latter's overall popularity. However, the average number of check-ins per user is higher in Gowalla: this could be due to the fact that our Foursquare dataset only contains a sample of user check-ins, namely those which were pushed to Twitter. The average number of check-ins per place, though, remains comparable across the two services.

When considering the entire temporal duration of the two datasets, 18 months in Gowalla and 5 months in Foursquare, there are about 10% of users and 25% of venues with only a single check-in in Gowalla; similarly, 20% of users and 35% of venues have a single check-in in Foursquare. This skew in the popularity of places and in user activity is reflected also
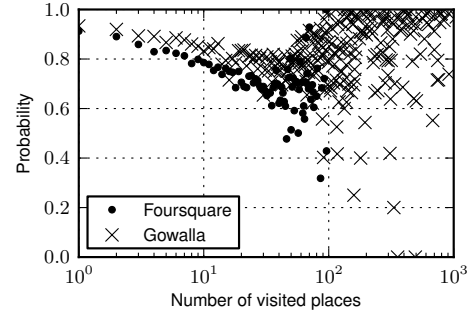


Fig. 3. Average probability of visiting a new place as a function of the number of places visited by users. A decreasing trend can be observed: this suggests that more active users are less likely to visit a new venue. The noise at the tail appears due to the low number of users with more than 50 check-ins.

in single cities. In fact, although each city exhibits different levels of user activity, the normalized distributions of check-ins across users and places are strikingly similar, as reported in Figure 1(a) and Figure 1(b). When each distribution is normalized by dividing each sample by the average value, all distributions collapse to a similar heavy-tailed pattern. In particular, about 80% of users have fewer check-ins than the average across all the cities, both in Gowalla and Foursquare. Similar patterns appear when considering the normalized distribution of check-ins at each place, presented in Figure 1(c) and Figure 1(d). Overall, the tail of each distribution reaches values of up to thousands of check-ins: a bulk of users with low activity coexists with a few extremely active users.

In Figure 2 we consider two monthly samples of the data, taken over two consecutive months $t$ and $t+1$; we then define as $\Psi_i^t = \Theta_i^{t+1} \setminus \Theta_i^t$ the set of new places visited by user $i$ in sample $t+1$. Then, we compute for each city across Foursquare and Gowalla two quantities: the probability $P_v$ that a visited venue was not previously visited

$$P_v = \frac{\sum\limits_i \|\Psi_i^t\|}{\sum\limits_i \|\Theta_i^{t+1}\|} \qquad (1)$$

and the probability $P_c$ that a check-in takes place in one of these new places defined as:

$$P_c = \frac{\sum\limits_i \sum\limits_{j \in \Psi_i^t} c_{ij}}{\sum\limits_i \sum\limits_{j \in \Theta_i^{t+1}} c_{ij}} \qquad (2)$$

Between 80% and 90% of visited places are *new* places, while between 60% and 80% of check-ins happen in these new venues. This demonstrates how *recommending new, unvisited places to users has an important value, as they often seek to discover new locations*. More in detail, we explore how this probability changes for users with a different number of visited places in Figure 3. Users that have a history of 10 visited places or less, have 80% probability of visiting a new place. However, as we consider more active users, which have checked in to several places over the last 30 days, this fractions

drops significantly. We proceed next by formalizing the task of recommending new venues to users as a prediction problem.

## IV. New Venue Recommendation

We begin by introducing the problem of new venue recommendation (Section IV-A). We then describe a number of algorithms that are suitable for the task, with a particular focus on the assumptions that they make about human mobility (Section IV-B): popularity baselines capture herding behavior; content-based filters assume that people will only be interested in a small set of venue categories; nearest-neighbor and matrix factorization-based collaborative filtering compute recommendations under the like-mindedness assumption; social filters model users exclusively based on their friends; lastly, spatial-filtering, by pruning candidates on physical distance, is tailored towards those who will not venture outside of a pre-defined geographic space.

### A. Problem Formulation

We formally define the *new venue recommendation problem* as follows: given a sample of check-in data taken over a time period $t$, a set of users $U$ and their check-ins across a set of venues $L$, we aim to predict the values of the set $\Psi_i^t = \Theta_i^{t+1} \setminus \Theta_i^t$, that is, the set of *new* places visited by each user $i$ in next time period $(t+1)$. Thus, we couple a training data set to a test data set which belongs to the following and non-overlapping temporal period. Note that we only predict check-in values for locations and users that have already previously appeared in our data at least once.

### B. Recommendation Algorithms: Assumptions about Mobility

In this section, we describe the set of algorithms that we examined for this prediction problem; the following section will outline our random walk-based method.

*1) Visiting Popular Venues:* The first (non-personalized) baseline ranks each user's unvisited venues based on their historical popularity: the `popularity` score $\hat{r}_k$ of place $k$ is computed as:

$$\hat{r}_k = \sum_{i \in U} c_{ik} \qquad (3)$$

In doing so, this method assumes that the likelihood of checking in is proportional to how many people have checked in before; **users will check in at the most popular places**. Recall, in fact, that Figure 1 showed the highly skewed distributions of user check-ins at venues; there are few venues that receive the majority of the check-ins, while many places remain relatively unvisited.

*2) Attending Venues by Category:* The next method is a content-based filtering approach [15]. The Foursquare data contains 313 place categories[5], whereas the Gowalla data contains 293. Given a user, we rank all the categories based on the number of check-ins made by the user in venues of each kind. Then, we populate a list of recommendations by ranking

[5]https://developer.foursquare.com/docs/venues/categories

unvisited venues according to their category; within-category venues are further ordered by their popularity. The underlying assumption is that **user preferences can be captured in a succinct group of categories**. This method further differs from global popularity by taking a first step into learning from user preferences: for example, users who frequently visits coffee shops will be recommended the most popular coffee shop, rather than the most popular venue in the city.

*3) Following Friends:* The availability of users' social ties allows for the possibility of recommending those venues visited by friends. The social filtering approach we consider ranks venues by summing the number of check-ins performed by a user's friends at each place. Formally, the `socialnet` score for a user venue pair is:

$$\hat{r}_{ik} = \sum_{j \in \Gamma_i} c_{jk} \qquad (4)$$

which operates solely based on user $i$'s set of friends $\Gamma_i$ check-ins to place $k$. This approach is based on the assumption that **users will exclusively visit the places visited by their friends** and builds on research exploring the interplay between human mobility and social factors[5], [22], [24]; the discovery of new events will thus propagate socially.

*4) Staying Close to Home:* Previous work has suggested that the *home* location of a user may constitute a good predictor of mobility and social event attendance [18]. Since we do not know the exact location where users live, we set their "home" location to the venue where they check in most frequently; we then rank potential new venues at increasing `distance` from the identified home. Although we may not infer their actual home location, this method assumes that **capturing the locality that users tend to frequently visit will increase the likelihood of finding new venues**. In other words, people will go to places near those that they already visit.

*5) Like-Mindedness and Similarity:* Collaborative Filtering (CF) has, to date, been the focal point of recommender system algorithm research [1]. Recommendations are computed based on the assumption that **historically like-minded users will continue to have shared preferences in the future**. Users are represented as a vector of check-ins that they have historically made to places, and items are viewed as a set of check-ins by users. In other words, these techniques assume that all the important information (both relating to preference as well as spatial dependencies) will be captured in the check-in frequency data. There are three approaches that we consider here: a user-based $k$-Nearest Neighbor, a item-based approach (which we denote `placenet`), and matrix factorization based on the Singular Value Decomposition of the user-venue check-in data.

User-based $k$NN directly compares user profiles to quantify the extent to which pairs of users check-in at the same venues. We measure the similarity $s_{ij}$ between a pair of users $i$ and $j$ based on the cosine-similarity of each users' check-in vectors. With this similarity matrix at hand, we can compute `kNN` recommendations for each user. Given a user $i$ we compute

a prediction score $\hat{r}_{ij}$ for place $j$ as the sum of a baseline estimate and weighted mean of normalized check-ins to that venue by similar users:

$$\hat{r}_{ij} = \frac{\bar{c}_j}{|\Phi_j|} + \frac{\sum\limits_{n \in U} ((c_{nj} - \bar{c}_n) \times s_{in})}{\sum\limits_{n \in U} s_{in}} \qquad (5)$$

The baseline estimate is the average check-ins to venue $j$ ($\bar{c}_j$) divided by the number of unique users to have visited place $j$ ($|\Theta_j|$); i.e., the average number of check-ins per user to that venue. Neighbor check-ins are first normalized by subtracting each user's mean check-ins, ($c_{nj} - \bar{c}_n$), and then weighted by the shared similarity with user $i$.

An alternative neighbor-based approach is to compute similarity across pairs of venues (rather than users) [20]: this variation captures the complementary assumption that **places are similar if visited by the same users**. To capture this idea in the context of new venue recommendations, our aim is to *connect* two places when they are visited by the same users and assign a weight to this connection by considering the number of distinct users that visit both. We thus form a graph, that we denote the `placenet`, whose nodes are places and the edge weight $p_{jk}$ between places $j$ and $k$ is defined as:

$$p_{jk} = |\Phi_j \cap \Phi_k| \qquad (6)$$

This graph allows us to rank a place $j$ according to the sum of the weights that connect it with the set of places visited by a user $i$:

$$\hat{r}_{ij} = \sum_{k \in \Theta_i} p_{kj} \qquad (7)$$

Finally, we also examine the effectiveness of using a CF algorithm based on a latent factor model `MF`. We represent the relationship between users and places as a matrix $R$, whose dimensionality is $N \times M$: that is, each row represents a user and each column represents a place, with $r_{ij} = c_{ij}$. This method maps both users and places to a joint latent factor space of dimensionality $F \ll N, M$, such that check-ins are modeled as inner products between vectors in that space. User $i$ is associated to a row vector $\mathbf{p}_i \in \mathbb{R}^F$ and place $j$ is associated with a column vector $\mathbf{q}_j \in \mathbb{R}^F$. The estimate for the number of check-ins made by user $i$ at place $j$ is thus $\hat{r}_{ij} = \mathbf{p}_i \mathbf{q}_j$. We learn the mapping from users and places to latent vectors by minimizing the regularized squared error $E$ over all the existing check-ins:

$$E = \sum_{i \in U} \sum_{j \in \Theta_i} (c_{ij} - \mathbf{p}_i \mathbf{q}_j)^2 + \lambda(\|\mathbf{p}_i\|^2 + \|\mathbf{q}_j\|^2) \qquad (8)$$

where the constant $\lambda$ regularizes the learned parameters, whose magnitudes are penalized. We adopt a stochastic gradient descent optimization algorithm to minimize the error [8]. In our implementation we set $F = 20$, since we have found this value to be a reasonable trade-off between scalability and accuracy: higher values of $F$ provide only diminishing returns.

## V. A RANDOM WALK APPROACH

Each method that we have presented above leverages one unique aspect of the data: for example, CF approaches capture like-mindedness and venue similarity, social filtering computes on friends' data, and spatial filtering solely considers physical distance. In this section we aim to achieve a better recommendation quality with an approach that can automatically combine each of these features: we define a network which connect places and users and we perform personalized random walks with restart to compute recommendations for individual users.

### A. Random Walk Models

A random walk over a linked structure is based on the idea that the connections between items encode information able to rank them in a useful way; as the random walker jumps across the graph's nodes according to transition probabilities, it will spend a different amount of time on each node: under certain assumptions, the random walk will approach a steady-state, resulting in a vector of steady-state probabilities for each node. These probabilities represent the desired output of a random walk model and are a function of both the structure of the network and of the transition probabilities assigned to links. A notable example in this domain is the use of PageRank [14] to rank Web pages. Personalized versions of PageRank have been designed in order to alter the ranking according to other factors, such as the topic of a page [10] or users' preferences. More generally, a *random walk with restart* can be adopted to personalize rankings: at any step there is a constant probability of jumping back to a target node, thus nodes that are closer to the target tend to be ranked more highly than distant nodes, providing a personalized view of the network [23].

In a random walk over a network, the transition probabilities can be arranged in a matrix $Q = \alpha W + (1 - \alpha)R$, formed by two factors, a structural one and a random one: $W$ encodes the transition probabilities according to the network structure, whereas $R$ models a random probability of jumping to any other node. The parameter $\alpha$ is used to tune the behavior[6]. The steady-state probability of node $i$ is $p_i$, and the steady-state probability vector $\mathbf{p}$ can be defined as the solution of the matrix equation $\mathbf{p} = \mathbf{p}Q$. A popular approach to compute $\mathbf{p}$ is to repeatedly iterate this equation until the vector converges, exploiting the sparsity of $Q$ to reduce memory requirements.

### B. Recommending with Random Walks

We represent the data that we have as an undirected graph whose nodes are users and venues. A user $i$ is linked to venue $j$ if $c_{ij}$ is non-zero; furthermore, a user is linked to another user if the pair are friends. This graph is used to define the structural transition matrix $W$ which contains a uniform transition probability for each edge. For every user $i$ we define a *random walk with restart*: at every step there is a constant probability of jumping back to the node of the user. In each case, the matrix $R$ encodes the probability of

---

[6]It is usually set to $\alpha = 0.85$ [14].

randomly jumping back from every node to the node of the user. In order to compute predictions for user $i$ we compute the steady-state probabilities of the related random walk: then we rank each place in decreasing order of steady-state probabilities. This favors places that are more connected, by any means, to the user: through friends, through visited places or through any combination of factors. The restart probability maintains the random walk in the user's neighborhood, thus biasing recommendation results towards venues that are more connected, in any sense, to the user. This simultaneously promotes places with several connections (i.e., popular) that are also reachable through friends and through already visited places. This feature is noted as `rwr`.

### C. Weighted Version

We also introduce a *weighted and directed* version of the random walk approach, denoted as `wrwr`, where each link is annotated with a weight that biases the transition probabilities, rather than having uniform probabilities on all the links going out from a node. Further, weights between two nodes can be different in opposite directions. A link from user $i$ to user $j$ is weighted as $\frac{1}{|\Gamma_i|}$; each friend of user $i$ is given an equal weight, that is inversely proportional to the total number of friends. A link from user $i$ to place $k$ is weighted as $\frac{c_{ik}}{\|\Theta_i\|}$, or proportionally to the user's check-ins to that venue over the total number of check-ins for that user. Finally, a link from place $k$ to user $i$ is weighted as $\frac{c_{ik}}{\|\Phi_k\|}$, or the user's check-in frequency at that place over the total number of check-ins for that place. All weights are normalized so that they represent transition probabilities: this is achieved by computing the sum of all the weights on the links going out from a node and then dividing every weight by this value.

## VI. EVALUATION

We now evaluate the recommendation algorithms and compare results across predictors, datasets and cities. In Section VI-A, we describe our experimental methodology and the three metrics that we use to evaluate recommendation quality. The results in Section VI-B then show that the sole method to outperform a popularity-based baseline is the Random Walk approach; finally, we discuss the implications of our results in Section VI-C.

### A. Methodology and Metrics

We partition the check-in data temporally into multiple training/test splits (each consisting of 30 consecutive days) in order to obtain cross-validated results. We then filter any check-in $c_{ij}$ from each test set if the training set has a non-zero entry for $c_{ij}$, i.e., if the user has already visited the venue. We note that users with no check-ins in the test set are not included in the performance evaluation. The output of each prediction algorithm is a per-user personalized ranked list of venues.

We use three metrics to quantify the quality of these recommendations. The first two, precision@N and recall@N, convert the outcome of each predictor into binary values: either

| Method | APR | Precision@10 | Recall@10 |
|---|---|---|---|
| Random | 0.500 | 0.000 | 0.003 |
| Popular | **0.228** | **0.026** | **0.089** |
| Activity | **0.228** | 0.025 | 0.087 |
| Home | 0.383 | 0.008 | 0.026 |
| Social | 0.392 | 0.015 | 0.049 |
| kNN | 0.443 | 0.003 | 0.011 |
| PlaceNet | 0.337 | **0.026** | **0.077** |
| MF | 0.281 | 0.004 | 0.014 |
| RW | **0.217** | **0.028** | **0.094** |
| Weighted-RW | **0.229** | 0.025 | **0.088** |

TABLE III
FOURSQUARE RESULTS: AVERAGE APR, PRECISION@10 AND RECALL@10 RESULTS.

| Method | APR | Precision@10 | Recall@10 |
|---|---|---|---|
| Random | 0.499 | 0.002 | 0.001 |
| Popular | **0.278** | **0.043** | **0.090** |
| Activity | **0.280** | 0.032 | 0.073 |
| Home | 0.341 | 0.023 | 0.042 |
| Social | 0.418 | 0.029 | 0.054 |
| kNN | 0.425 | 0.005 | 0.012 |
| PlaceNet | 0.338 | **0.043** | **0.077** |
| MF | 0.343 | 0.009 | 0.025 |
| RW | **0.232** | **0.048** | **0.095** |
| Weighted-RW | **0.244** | **0.045** | **0.095** |

TABLE IV
GOWALLA RESULTS: AVERAGE APR, PRECISION@10 AND RECALL@10 RESULTS

the user will visit the top-N venues and will not visit venues ranked below N. Precision (p) and recall (r) are measured as proportions of true positives (tp), false positives (fp), and false negatives (fn):

$$p = \frac{tp}{tp + fp}; r = \frac{tp}{tp + fn} \qquad (9)$$

Since each algorithm outputs an ordered list, we also verify the extent to which the ranking reflects users' interests; i.e., that venues that are highly ranked are indeed those that will be more frequently visited. To do so, we first define the *interest* $i$ that a user $u$ has in a venue $s$ as the proportion of times that the user checks into that venue during the test period. We then define $rank_{u,s}$ as the percentile ranking of venue $s$ for user $u$ in the ranked list of venues; if $rank_{u,s} = 0$, then the venue appears first in the list, while $rank_{u,s} = 1$ implies that the venue was the last in the list. We combine these with each user's interest in the station $interest_{u,s}$ and average the results to measure the Average Percentile Ranking (APR):

$$\overline{APR} = \frac{\sum_{u \in U} \sum_{s \in L} i_{u,s} \times rank_{u,s}}{\sum_{u \in U} \sum_{s \in L} i_{u,s}} \qquad (10)$$

In the following sections, we report and discuss the empirical results we obtained following the above methodology and metrics.

| | Foursquare | | Gowalla | |
|---|---|---|---|---|
| **City** | popularity | rwr | popularity | rwr |
| Austin | 0.235 | 0.222 | **0.175** | **0.144** |
| Boston | 0.204 | 0.196 | 0.313 | 0.258 |
| Dallas | 0.247 | 0.232 | 0.248 | 0.197 |
| Denver | 0.233 | 0.200 | 0.285 | 0.236 |
| London | 0.264 | 0.262 | 0.311 | 0.244 |
| Los Angeles | 0.212 | 0.196 | 0.281 | 0.242 |
| New York | **0.192** | **0.185** | 0.280 | 0.242 |
| Paris | 0.265 | 0.256 | 0.271 | 0.204 |
| San Francisco | 0.208 | 0.200 | 0.220 | 0.183 |
| Seattle | 0.238 | 0.218 | 0.264 | 0.226 |
| Seoul | 0.210 | 0.226 | 0.410 | 0.381 |
| **Average** | 0.228 | 0.217 | 0.278 | 0.232 |

TABLE V

APR ACHIEVED BY THE POPULARITY AND RWR PREDICTION METHODS ACROSS CITIES IN BOTH DATASETS. FOR EACH SERVICE WE HIGHLIGHT THE CITY WITH THE BEST APR SCORE.

### B. Results

In order to put the following results into an appropriate context, we also compare them to a random predictor, which simply shuffles the candidate set of unvisited venues for each user. In this case, APR results are 0.5 and both precision and recall are near zero. We further note that better results are obtained with *lower* APR values and *higher* precision and recall values.

*1) Performance Across Methods:* Tables III and IV show the APR, precision@10 and recall@10 for the Foursquare and Gowalla datasets respectively. The most eminent result is that nearly all methods, including social filtering and all (kNN and MF) versions of collaborative filtering—which were supposed to better model users' preferences— fail to outperform the popularity-based baseline. The Activity predictor, which ranks venues based on the categories of the venues visited by each user, also ranks amongst the top performing approaches. The random walk variants are the only approaches that outperform popularity, and are amongst the top performing methods for both datasets: rwr achieves an improvement of 5% in Foursquare and of 18% in Gowalla with respect to popularity, the best performing among the other methods.

In general, the three metrics agree with each other in terms of algorithms' relative performance, with one exception. When considering precision and recall, both placenet and, to a lesser extent, socialnet achieve results similar to the best four methods. This difference between APR and precision/recall is due to the fact that placenet and socialnet are the only two methods that do not rank all the available places, but only a subset of places specific to the target user. As their recommendation lists may thus contain fewer items, they are penalized in the APR score, which is agnostic to list size, but they benefit in precision and recall, where list size is important.

*2) Performance Across Cities:* Although there are performance variations across cities, our method based on a random walk consistently outperforms the other approaches. This observation is reflected in Table V, which presents the APR
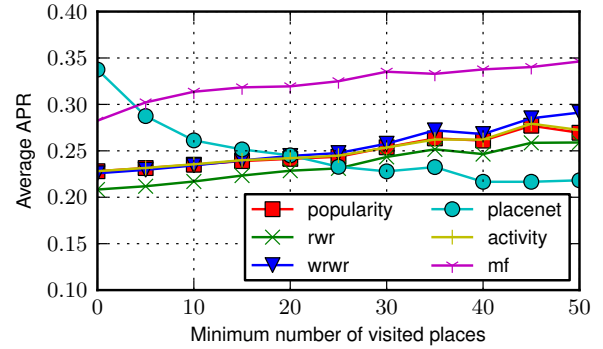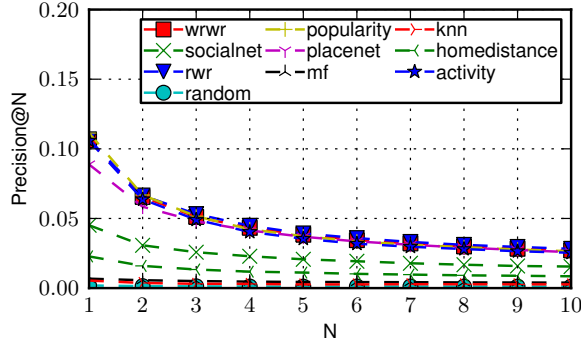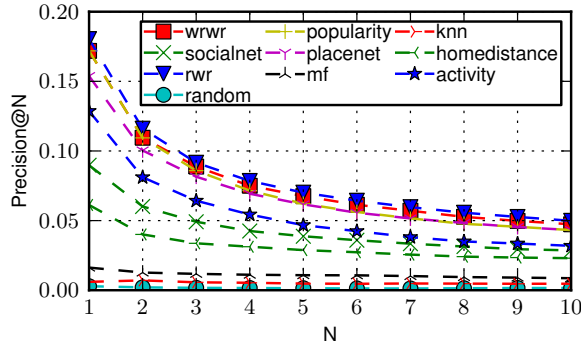


(a) Foursquare
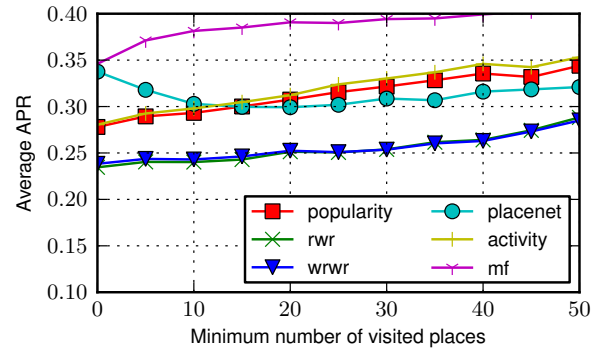


(b) Gowalla

Fig. 4. Average APR of the best approaches for users with a different minimum number of visited places in the training snapshot in Foursquare (a) and Gowalla (b).

scores for popularity and rwr achieved across different cities in both services; we observed a similar pattern for the precision and recall metrics (not shown). Moreover, this analysis suggests that there is no strong correlation between the individual city statistics presented in Tables I and II, such as the number of active users and places, and the prediction performance. One outlier seems to be Seoul in Gowalla, with much lower performance: this might be due to the fact that this case has the lowest level of user activity across all the considered cities. On the other hand, New York in Foursquare and Austin in Gowalla, which are the cities with most check-ins in the corresponding datasets, show the best performance.

Furthermore, the results obtained across the two datasets agree with one another; this is a notable result since both systems have different interfaces and incentives for user participation. We must note that no location recommendation engine was put in place by either service during the data collection period. The most prominent difference in performance is obtained with the home distance feature, that is consistently better in Gowalla than in Foursquare across all the performance metrics. This could be due to the fact that the average number of user check-ins in Gowalla is higher than that observed in Foursquare, thus allowing the "home" location inference to be more accurate. In addition, in Gowalla,
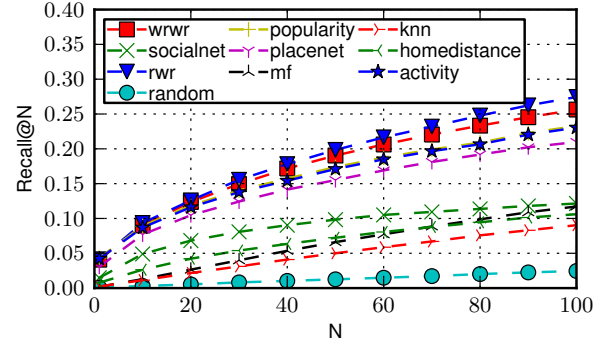
(a) Foursquare



(a) Foursquare



(b) Gowalla



(b) Gowalla

Fig. 5. Average Precision@N obtained by each algorithm on all cities for various values of $N$ in Foursquare (a) and Gowalla (b).

Fig. 6. Average Recall@N obtained by each algorithm on all cities for various values of $N$ in Foursquare (a) and Gowalla (b).
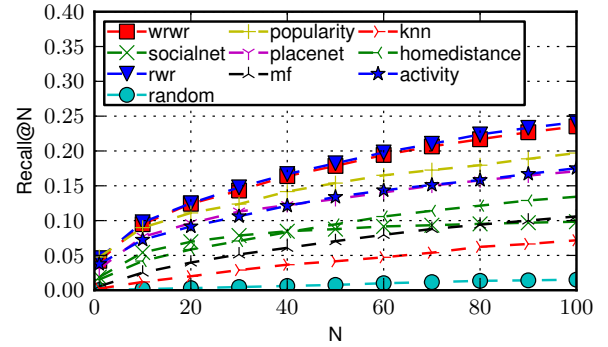
when the entirety of user social links and check-ins are present, the random walk models achieve a larger performance gain, as they are able to exploit higher quality data to build the network structure.

*3) Impact of User Activity:* We have discussed how users which have visited more places tend to visit fewer new places, as presented in Figure 3. Thus, we investigate how prediction performance changes when we consider users with progressively higher amount of visited places. We filter out users which have visited less than a certain number of places in the train snapshot and we compute how the average APR over all the remaining users changes when we progressively increase the minimum number of visited places. As shown in Figure 4, where we depict some of the best performing methods, prediction performance decrease as we focus on the most active users. A noticeable difference is the `placenet` feature, which achieves better results when we filter out less active users, while overall its performance hardly compete with the best methods. However, as the vast majority of users have visited only few places, any improvement for active users is not likely to impact the performance across the entire user base.

*4) Impact of Recommendation List Size:* Finally, we explore the effect of a varying recommendation list size on the final performance. As the number of venues recommended to a user increases recall tends to improve, while precision

might suffer. In Figure 5 we plot the average precision@N obtained by each algorithm on all cities: we find that precision decreases as we increase list size, with the different features maintaining their relative ranking in performance. Similarly but with an opposite trend, recall@N quickly increases as the recommendation list grows larger, as depicted in Figure 6; again, the dominating features outperform the others over the entire range of list size. This analysis highlights the trade-off between precision and recall that each feature faces: real systems should tune their results according to what users require.

*C. Discussion*

The data that we obtained has a number of characteristics that differentiate it from the typical recommender system scenario. An important difference is that while in other scenarios users reveal their preferences through ordinal ratings, in our case check-ins only capture numeric frequencies: as a consequence, there is no negative feedback provided by users. Furthermore, the data is highly sparse, with many users and venues having a single check-in. At the same time, across both datasets there are few places with extremely high number of check-ins, while the majority of them only enjoys few user check-ins. Thus, there is a high heterogeneity across how users check in at places, with some venues reaching high levels of popularity. This may go some way to explaining the high

performance of popularity-based recommendations.

The previous section also uncovered that the worst performing algorithms for this task were the ones that are typically associated with recommender systems (i.e., nearest neighbors and matrix factorization). There are a number of reasons why this may be the case. First, check-in data may not be sufficient to fully capture users' preferences. In fact, unlike web ratings, it leans towards capturing habitual behavior well and does not allow for negative feedback. Second, like-mindedness may not suffice to model why people visit venues; the random-walk approaches outperform standard CF by simultaneously leveraging several sources of data, encoding them in the network structure. Moreover, computing recommendations using a random walk is not tied to this particular dataset: it is generalizable to any situation where signals of user preference can be encoded into a graph.

## VII. Conclusion

This paper has examined the novel problem of learning to predict which *previously unvisited* places users of location-based social networks will visit, in order to build a new-venue recommendation system. We collected behavioral, social, and spatial data from Gowalla and Foursquare (via Twitter) for a range of the world's metropolises, and evaluated how a variety of learning algorithms—from simple, non-personalized popular venue recommendation to predictions based on matrix factorization methods—were able to rank and classify the new venues for each user. We found that the collaborative filtering approaches that have been successful in online recommendation scenarios have not achieved a similar status with mobility data: instead, we found that our proposed random-walk based model was able to consistently achieve the best performance, by learning from both social ties and venue-visit data simultaneously.

The problem setting that we have examined captures a characteristic that all recommender systems will soon face: the abundance of social-based and contextual data beyond explicit ratings that is available to improve users' recommendations. For example, even services such as Netflix (which have been at the focal point of recommender system literature since the Netflix Prize) are turning to streaming films and connecting their services to online social networks like Facebook. This new generation of services, often mainly used on mobile devices, are likely to already hold social, behavioral, and geographical information about their users and the mobile devices that are used to stream films. The open research challenge is thus to design learning algorithms that can leverage all this information for better recommendations.

In terms of future work, we seek to evaluate supervised learning methods that could be accommodated in the new venue recommendation problem. Supervised frameworks have been applied successfully to social based systems [22]: even social link prediction systems based on random walks have benefited from the use of supervised learning, as shown by the authors in [2]. Finally, matrix factorization techniques could be extended to include the multiple sources of information available in check-in datasets of location-based social services.

## References

[1] G. Adomavicius and A. Tuzhilin. Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE TKDE*, 17(6):734–749, June 2005.

[2] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of WSDM '11*, pages 635–644, New York, NY, USA, 2011. ACM.

[3] R. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Proceedings of ICDM '07*, pages 43–52. IEEE, 2007.

[4] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of KDD '11*, pages 1082–1090, New York, NY, USA, 2011. ACM.

[5] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of KDD '11*, pages 1082–1090, New York, NY, USA, 2011. ACM.

[6] J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh. Bridging the Gap Between Physical Location and Online Social Networks. In *Proceedings of UBICOMP '10*, pages 119–128, New York, NY, USA, 2010. ACM.

[7] N. Eagle, A. S. Pentland, and D. Lazer. Inferring friendship network structure by using mobile phone data. *PNAS*, 106(36):15274–15278, September 2009.

[8] S. Funk. Netflix Update: Try This At Home. http://sifter.org/~simon/journal/20061211.html, 2006.

[9] J. Golbeck. *Computing With Social Trust*. Springer, 2008.

[10] T. H. Haveliwala. Topic-sensitive pagerank. In *Proceedings of WWW '02*, pages 517–526, New York, NY, USA, 2002. ACM.

[11] Y. Koren. Collaborative Filtering with Temporal Dynamics. In *Proceedings of KDD '09*, pages 89–97, Paris, France, 2009.

[12] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. Wherenext: a location predictor on trajectory pattern mining. In *Proceedings of KDD '09*, pages 637–646, New York, NY, USA, 2009. ACM.

[13] A. J. Nicholson and B. D. Noble. BreadCrumbs: Forecasting Mobile Connectivity. In *Proceedings of MobiCom '08*, pages 46–57, New York, USA, 2008.

[14] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.

[15] M. Pazzani and D. Billsus. Content-Based Recommendation Systems. *The Adaptive Web*, 4321:325–341, 2007.

[16] A. Popescu and G. Grefenstette. Deducing Trip Related Information from Flickr. In *Proceedings of WWW '09*, Madrid, Spain, April 2009.

[17] D. Quercia and L. Capra. Friendsensing: recommending friends using mobile phones. In *Proceedings of RecSys '09*, pages 273–276, New York, NY, USA, 2009. ACM.

[18] D. Quercia, N. Lathia, F. Calabrese, G. Di Lorenzo, and J. Crowcroft. Recommending social events from mobile phone location data. In *Proceedings of ICDM '10*, pages 971 –976, dec. 2010.

[19] A. Sadilek, H. Kautz, and J. P. Bigham. Finding your friends and following them to where you are. In *Proceedings of WSDM '12*, pages 723–732, New York, NY, USA, 2012. ACM.

[20] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of WWW '01*, pages 285–295, New York, NY, USA, 2001. ACM.

[21] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell. NextPlace: a spatio-temporal prediction framework for pervasive systems. In *Proceedings of Pervasive '11*, pages 152–169, Berlin, Heidelberg, 2011. Springer-Verlag.

[22] S. Scellato, A. Noulas, and C. Mascolo. Exploiting place features in link prediction on location-based social networks. In *Proceedings of KDD '11*, New York, NY, USA, 2011. ACM.

[23] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. In *Proceedings of ICDM '06*, pages 613–622, Washington, DC, USA, 2006. IEEE Computer Society.

[24] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A. L. Barabasi. Human mobility, social ties, and link prediction. In *Proceedings of KDD '11*, pages 1100–1108, New York, NY, USA, 2011. ACM.

[25] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of WWW '09*, pages 791–800, New York, NY, USA, 2009. ACM.