# A personalized geographic-based diffusion model for location recommendations in LBSN

Iury Nunes
*Federal University of Campina Grande*
*Campina Grande, Brazil*
*iury.nunes@ccc.ufcg.edu.br*

Leandro Marinho
*Federal University of Campina Grande*
*Campina Grande, Brazil*
*lbmarinho@dsc.ufcg.edu.br*

***Abstract*—Location Based Social Networks (LBSN) have emerged with the purpose of allowing users to share their visited locations with their friends. Foursquare, for instance, is a popular LBSN where users endorse and share tips about visited locations. In order to improve the experience of LBSN users, simple recommender services, typically based on geographical proximity, are usually provided. The state-of-the-art location recommenders in LBSN are based on linear combinations of collaborative filtering, geo and social-aware recommenders, which implies fine tuning and running three (or more) separate algorithms for each recommendation request. In this paper, we present a new location recommender that integrates collaborative filtering and geographic information into one single diffusion-based recommendation model. The idea is to learn a personalized ranking of locations for a target user considering the locations visited by similar users, the distances between visited and non visited locations and the regions he prefers to visit. We conduct experiments on real data from two different LBSN, namely, Gowalla and Foursquare, and show that our approach outperforms the state-of-art in most of the cities evaluated.**

***Keywords*-Recommender Systems, Location Based Social Networks, Collaborative Filtering, Diffusion Model, Location-Aware**

## I. Introduction

Location-based Social Networks (LBSN) or Geosocial Networks are online social networks in which geographic services and capabilities are used to let users check-in and then broadcast their locations. Prominent examples of LBSN are Foursquare[1] and Facebook places[2] whose main goal is to help people meet up with friends and discover interesting places. Thanks to the affordable prices of GPS enabled smartphones and other mobile devices, these services are now widely spread and poised for continued growth. Being released in 2009, Foursquare, for example, reached over 50 million users and 6 billion check-ins worldwide in 2014[3]. When the space of choices available (which places to visit?) gets too large, the problem known as information overload emerges.

[1]http://foursquare.com
[2]http://www.facebook.com/about/location
[3]https://foursquare.com/about

Recommender systems appear as a natural solution to this problem offering effective techniques for helping users to filter out and discover relevant information in large data sets. Foursquare uses the current location of users to recommend nearby locations based on existing categories. For example, users might want to know which restaurants he can visit nearby his current location. Although this certainly helps users to reduce the space of choices available, it does not take into account their personal preferences. Users might be interested, for example, in nearby restaurants within a certain price range or that offer live music. In fact, the state-of-the-art works on location or points-of-interest (POI) recommendations propose a combination of collaborative filtering (for capturing personal POIs preferences of users) and some recommender that captures the preferences of users for geographic regions (aka geographic-aware recommender). In many cases, a social-aware recommender is also included in the ensemble [3, 17, 18]. Although these algorithms have presented promising results, they require fine tuning and execution of two or more specialized algorithms for each recommendation request. Moreover, the personal preferences of users for POIs and geographic regions are modeled separately, while there is a clear correlation between them, e.g., the preference of a user for a region may reinforce the preferences of this user for the POIs in that region and vice-versa.

In this work, we propose to model the preferences of users for locations, the distances between visited and non visited locations, and the preferences of users for geographic regions in a weighted graph. We then use a random walk algorithm for capturing, in a transparent way, the interplay between these three kinds of user preferences. We conduct experiments on real data from two different LBSN, namely, Gowalla and Foursquare, and show that our approach outperforms the state-of-art in most of the cities evaluated. Our contributions are as follows:

- We propose a new model for check-in data that is able to capture, at the same structure, the distances between locations and the preferences of users for POIs and geographic regions.

978-1-4799-6953-1/14 $31.00 © 2014 IEEE
DOI 10.1109/LAWeb.2014.22

- We propose to exploit the interplay between these three kinds of data through a diffusion model, i.e., a random walk algorithm.
- We evaluate our approach on two large-scale data sets and show that our approach outperforms the state-of-the-art algorithms in most of the cities evaluated.

The rest of the paper is organized as follows. In Section II we formalize the problem approached by this paper. In Section III we present the related work and position this paper among them. In section IV we recall the basic concepts of the random walk technique. In Section V we describe our approach in detail. In Section VI we present the evaluation protocol used and discuss the outcome of the experiments. Finally, Section VII concludes the paper and discusses the outlook.

## II. Problem Setting

The recommendation scenario that we investigate in this paper is as follows: a user specifies the city of interest and the recommender engine suggests locations (or POIs) within the selected city that are likely to be relevant to the user. It is important to remark that the city of interest might be either the user's hometown or a city where the user is traveling. The only requirement is that the user must have at least one check-in in the city of interest.

Thus, let $U$ be the set of users, $L$ the set of locations and $C$ the set of cities. In this paper we only consider implicit feedback data, i.e., the set $S \subseteq U \times L \times C$ of ternary relations between users and geotagged venues. The task is then to find a scoring function

$$\hat{s} : U \times L \times C \rightarrow \mathbb{R} \tag{1}$$

that assigns a preference score for locations within a certain city, given a target user. Thus, for a given user $u \in U$, the topN recommendations can be computed by

$$\mathit{topN}(u, c) := \operatorname*{argmax}^{n}_{l \in L_c \setminus L_u} \hat{s}(u, l, c) \tag{2}$$

where $n$ denotes the number of locations to be recommended, $L_c$ the set of locations within city $c$ and $L_u$ denotes the set of locations checked-in by user $u$. For convenience, we also define $U_l$ as the set of users that checked-in at $l$.

## III. Related Work

Several research works have exploited check-in data of LBSN as a useful source of information for understanding human mobility patterns [5, 4, 12, 16]. Although these works are not directly related to recommender systems, the insights they provide can be used to devise novel and effective location-aware recommendations. For example, two important insights coming from some of these works are: (i) the distribution of distances between pairs of visited locations by a user resembles a powerlaw, i.e., the majority of locations that a given user has checked-in at are close to each other; and (ii) users tend to concentrate their check-ins around a few regions, e.g., their homes and/or workplaces.

This topic of research is closely related to context-aware recommendations, in which the context of interest is the geographic position of items and/or users. The geographic context is very challenging by itself [13], but other contexts were also investigated by the literature, like the timestamp [6] and the category of checked-in locations (e.g., "food", "museum", "stadium", etc.) [1, 11]. Although these contexts are indeed important and worth investigating, in this work we focus on the the geographic context given that this is the core feature of LBSN.

Most of the existing research work on location recommendation proposes some sort of combination between collaborative filtering and some recommender that exploits the geographic preferences of users. Cheng et al. [3] combined, through a simple multiplication, a probabilistic factor model for collaborative filtering with a Multi-center Gaussian Model for modelling the geographic preferences of the users. Ye et al. [17] in turn, fused, through a simple linear combination, neighborhood-based collaborative filtering with a power law based model for modelling the geographic preferences of users. We present more details about these two recommenders in Section VI-B. Instead of using different recommendation algorithms in an ensemble as the aforementioned works do, we define a single recommendation algorithm that takes into account both collaborative filtering and the geographic preferences of users in a transparent fashion.

More recently, location recommenders based on probabilistic topic models have appeared [7, 10, 18]. Differently from us, the works of [7, 10] do not use data from LBSN, but from services like yelp, twitter and flickr. Although there are geotagged data in these services, they do not convey the same information that check-in data does. For example, the fact that a given geotagged photo was shared in flickr does not imply that the user who uploaded the photo wanted to share the location where the photo was shot. Differently from these two works, [18] used data from LBSN and EBSN (Event Based Social Networks) to evaluate their recommendation model. Their approach proves to outperform the method of Ye et al. [17] in several scenarios, except in the scenario we investigate in this paper, i.e., recommending for a user who has some chek-ins in the city he is located and is looking for recommendations within that city. Hence, we will compare our approach with the approaches of [17] and [3].

Random walk-based recommendation models have proved to be very effective in domains other than location recommendation [8, 2, 9]. We complement these works introducing a new information diffusion model that captures, at the same time, collaborative filtering, the distance between POIs and the geographic preferences of users.

## IV. Random Walk on Graphs

Before we introduce our diffusion-based recommendation model in the next section, we briefly recall the basic principles of information propagation on graphs using random walk. A popular implementation of random walk is the PageRank algorithm, which exploits the hyperlink structure of web pages under the assumption that a web page is important if there are many pages linking to it, and if those pages are important themselves [14].

A graph is defined as a tuple $G = (V, E)$ where $V$ is a set of vertices (or nodes) and $E$ a set of edges $E \subseteq V \times V$. An edge $e_{u,v} \in E$ means that there is a link between $u \in V$ and $v \in V$. A graph can be weighted in order to denote that some edges are more important than others. Thus, let $w : E \to \mathbb{R}$ be the function that assigns weights to edges in $G$.

The idea behind random walk algorithms is that there is a "walker" visiting vertices randomly in the graph. At each iteration, the "walker" will jump from one node to another according to the weights of the edges leaving the current node. The larger the weight of an edge, the larger the probability of the "walker" passing through that edge. As the number of iterations increases, the "walker" will visit some nodes more often than others. The idea is that the nodes that were visited more frequently should be ranked higher because they are more important.

We can also represent a graph $G$ as an adjacency matrix $A \in \mathbb{R}^{|V|\times|V|}$ where the value at entry $A_{i,j}$ corresponds to $w(e_{i,j})$. If we model the weights of each edge $e_{i,j} \in E$ as the probability of the "walker" jumping from the node $i$ to node $j$ in the next iteration, then we can compute the probabilities of the user being at any node given a number of iterations $t$. As the number of iterations grows, these probabilities will converge to a steady state. So, let $\vec{x}$ be a vector with $|V|$ dimensions. At each iteration, this vector will be updated with a probability distribution of the nodes.

Since the probability distribution of $\vec{x}$ will eventually converge, it doesn't really matter in which node the walker starts. So, at iteration $t = 0$ we randomly choose any vertex $v \in V$ and assign the value 1 to it, i.e., $\vec{x}_0[v] = 1$ and 0 for all the other nodes $u \in \{V \setminus \{v\}\}$. Now, at each subsequent iteration we update the values of $\vec{x}$ as described in Equation 3 where $\vec{p}$ is a vector (typically filled with a uniform probability distribution) that can be used for asserting preferences for specific nodes and the teleport factor $0 < \lambda < 1$ is used for determining the strength of the influence of $\vec{p}$. In practice, the teleport factor refers to the probability of the "walker" jumping to any other node in the graph, even the nodes that are not linked to the current node. The teleport factor is important to increase the probability of visiting vertices that have only a few or no incoming edges. The process stops when there is no significant change between $\vec{x}_t$ and $\vec{x}_{t+1}$.

$$\vec{x}_{t+1} = \lambda A^T \vec{x}_t + (1 - \lambda)\vec{p} \tag{3}$$

Other versions of this algorithm also uses the concept of restart. While the teleport allows the "walker" to jump uniformly to any node, the restart factor will bias the jump towards the starting node. As we will see in Section V-A, we modeled each user as a node. Since the goal of the recommender is to generate personalized recommendations for a given target user, the node of such user should be considered more important than the other nodes. So, if we set the target user node as the starting node and increase the influence of the restart factor, we guarantee that the "walker" will visit this node more often. Thus, the steady state achieved by the random walk will be biased towards the target user personal preferences.

## V. A Diffusion Model for Location Recommendations

Inspired by the findings of related works, we designed a location recommender that takes the following assumptions into account:

1) Users that visited similar locations in the past tend to visit the similar locations in the future [17].
2) Users tend to visit locations close to the locations they have already visited in the past [4, 12, 13].
3) Users tend to concentrate their check-ins around a few regions of interest [5, 3].

In subsection V-A we present a graph representation for check-in data. Next, in subsections V-B, V-C and V-D we describe, step by step, how we can approach each one of these three assumptions using a graph-based diffusion model. Finally, in subsection V-E these three assumptions will be integrated into a single recommendation model.

### A. Modelling Check-in Data on a Graph

We structure the implicit feedback data of LBSN as a graph where nodes are comprised of users and locations, assuming that these users and locations belong to some given city. There is an edge between a node $u \in U$ and a node $l \in L$ if user $u$ has checked-in at $l$. If we define a Boolean function checkedIn$(u, l)$ to denote whether the user $u$ checked-in at location $l$, we can formally define graph $G$ as follows: $G = (V, E)$ where $V = U \cup L$ and $E = \{\{(u, l) : u \in U, l \in L, \text{checkedIn}(u, l) = \mathbf{true}\} \cup \{(l, u) : u \in U, l \in L, \text{checkedIn}(u, l) = \mathbf{true}\}\}$.

Figure 1 depicts an example of this graph where $U = \{u_1, u_2, u_3\}$ and $L = \{l_1, l_2, l_3, l_4, l_5\}$. Since one of the main goals of recommender systems is to help users finding new items (in our case locations) we will not consider the locations that the target user $u_1 \in U$ has already checked-in at. The locations $l_1$ and $l_3$ are colored in red in Figure 1 to denote that these locations have been already checked-in by

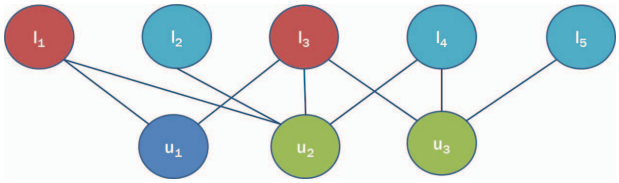


Figure 1: Graph depicting the relation between users and their checked-in locations.

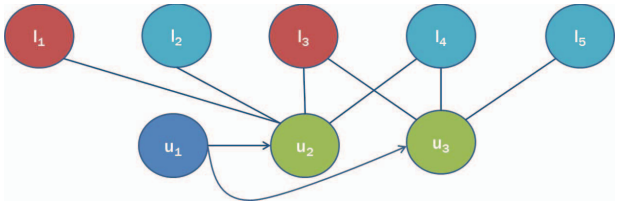


Figure 2: Collaborative Filtering Graph

the target user. Thus, the only recommendable locations for user $u_1$ are $l_2, l_4$ and $l_5$.

If we weigh each outgoing edge, for any node in $V$, with the same probability and perform a random walk over this weighted graph restricting the final probabilities to the nodes of type location, we would end up with a ranking that corresponds to the most popular locations, i.e., the locations that were checked-in more often by distinct users. More formally, let $\text{outDegree}(u) = |\{v \in V : (u,v) \in E\}|$ be the out degree of a given node $u \in V$ of the graph, then the weights of the graph take the form of Equation 4.

$$w(u,v) = \frac{1}{\text{outDegree}(u)} \tag{4}$$

Notice that the popularity-based recommender does not provide personalized recommendations so we will not consider it in our model. In the following subsection we will describe how to achieve personalization by means of collaborative filtering.

### B. Diffusion-based Collaborative Filtering

The collaborative filtering assumption is that users who shared the same interests in the past tend to share the same interests in the future. The user-based collaborative filtering (based on K-nearest neighbors) is a classic recommender that, despite its simplicity, has proven to attain high accuracy in LBSN [17].

To implement this algorithm we need to define a measure that captures the similarity between a pair of users. We have chosen to use the well known cosine similarity measure which have been successfully used in many domains including LBSN [17]. First, we define, for each user $u \in U$, a vector $\vec{u}^{|L|}$ whose components are equal to 1 if the user checked-in at the corresponding location and 0 otherwise. The cosine similarity between any two users $u, v \in U$ is then defined as follows:

$$sim^{cf}(u,v) = \frac{\vec{u} \cdot \vec{v}}{||\vec{u}||||\vec{v}||} \tag{5}$$

Now, for finding the $k$-nearest neighbors of a given target user $u \in U$ we only need to compute the similarity between $u$ and every other user and sort these users in descending order of similarity down to $k$.

Let $N_u$ be the set of $k$-nearest neighbors of target user $u$ and

$$L_{N_u} = \bigcup_{v \in N_u} L_v$$

be the set of all locations visited by all neighbors of $u$. Now, a user-based collaborative filtering graph can be defined as $G_{cf} = (V_{cf}, E_{cf})$ where $V_{cf} = \{u\} \cup N_u \cup L_{N_u}$, $E_{cf} = \{(u,v) : v \in N_u\} \cup \{(v,l) : v \in N_u, l \in L_{N_u}\} \cup \{(l,v) : l \in L_{N_u}, v \in N_u\}$ and the edge weights are defined as:

$$w^{cf}(p,q) = \begin{cases} \frac{sim^{cf}(p,q)}{\sum_{v \in N_u} sim^{cf}(p,v)}, & \text{if } p = u \text{ and } q \in N_u \\ \frac{1}{|L_p|}, & \text{if } p \in N_u \text{ and } q \in L_{N_u} \\ \frac{1}{\text{outDegree}(p)}, & \text{if } p \in L_{N_u} \text{ and } q \in N_u \\ 0, & \text{otherwise} \end{cases}$$

Notice that the values of the weights of the edges are normalized so that the sum of the weights of the outgoing edges of each node is not greater than 1. Figure 2 depicts how the graph of Figure 1 becomes a collaborative filtering graph assuming that $u_2 \in U$ and $u_3 \in U$ are the target user neighbors. In this example, the weights of each outgoing edge of $u_2$ would be 0.25 and the weights of the incoming edges of $u_2$ from $l_1$, $l_2$, $l_3$ and $l_4$ would be 1, 1, 0.5 and 0.5 respectively. Notice that if we apply a random walk on this graph, location $l_4$ would be ranked higher than $l_2$ and $l_5$ because $l_4$ can be reached through $u_2$ and $u_3$ whereas $l_2$ can only be reached through $u_2$ and $l_5$ can only be reached through $u_3$.

### C. Diffusion-based Pairwise Distances

It is not a surprise that the check-ins of the users are not uniformly distributed over the map. It generally demands time and money to visit locations which are far from users homes. Thus, people tend to visit locations that are close to the locations they already visited in the past [4, 12].

We capture this assumption in the graph $G_{dist}$, where edges between locations are created and weighted with the distances (in kilometers) between them. As a similarity measure between two locations $l$ and $l'$, we use the one defined in Equation 6, where $\text{dist}(l,l')$ computes the geographical distance between locations $l$ and $l'$.

$$sim^{\text{dist}}(l,l') = \min(1, \frac{1}{\text{dist}(l,l')}) \tag{6}$$

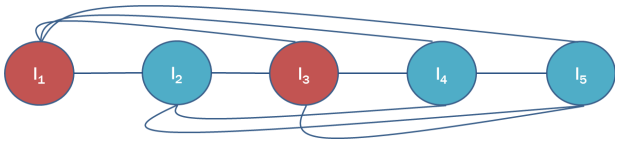


Figure 3: Distance Based Model Example

We use the function min, instead of $1/\text{dist}(l, l')$, in order to have a similarity measure in the range $[0, 1]$. If we used only the inverse of the distance as the similarity between these locations, we would have a similarity equal to 2 if the distance between the locations was equal to 0.5 kilometers, for example.

Now we can formally define $G_{\text{dist}}$ as follows: $G_{\text{dist}} = (V_{\text{dist}}, E_{\text{dist}})$ with $V_{\text{dist}} = L$, $E_{\text{dist}} = L \times L$ with the edge weights defined in Equation 7. Notice that the weights of the edges are simply the similarity measure defined in Equation 6 normalized.

$$w^{\text{dist}}(p, q) = \frac{sim^{\text{dist}}(p, q)}{\sum_{l \in V_{\text{dist}} \setminus \{p\}} sim^{\text{dist}}(p, l)} \tag{7}$$

Figure 3 depicts the locations presented in the previous figures but now with links between them. If we now perform a random walk on this graph assuming that all the locations are geographically disposed as presented in Figure 3 and that $\text{dist}(l_1, l_2) = \text{dist}(l_2, l_3) = \text{dist}(l_3, l_4) = \text{dist}(l_4, l_5)$, we would rank $l_2$ and $l_4$ on the top (both having the same ranking score) followed by $l_5$.

### D. Diffusion-based Regions of Interest

Users tend to check-in locations in a few well defined regions [5, 3]. For example, two typical regions where users tend to concentrate their check-ins are the regions around their homes and work.

In order to model this assumption we first need to infer the regions of interest for users since this is not given in the check-in data set. For that, we employed the same approach as [5], where the world is discretized into a grid so that each cell becomes a region. In our case, we considered 20 by 20km cells. Although in [5] it was used 25 by 25km cells, we achieved better results decreasing the size of the cell to 20km.

Let $R$ be the set of regions, $L_r$ the set of locations within region $r \in R$, and $L_{u,r}$ the set of checked-in locations user $u$ has done at region $r$. Now, for a given target user $u \in U$ we define a graph for modelling user preferences for regions as $G_{\text{reg}} = (V_{\text{reg}}, E_{\text{reg}})$ where $V_{\text{reg}} = \{u\} \cup R \cup L$, $E_{\text{reg}} = \{(u, r) : r \in R\} \cup \{(r, l) : r \in R, l \in L_r\}$ and the edge weights are defined as follows:

$$w^{\text{reg}}(p, q) = \begin{cases} \frac{|L_{p,q}|}{|L_p|}, & \text{if } p = u \text{ and } q \in R \\ \frac{1}{|L_p|}, & \text{if } p \in R \text{ and } q \in L_p \\ 0, & \text{otherwise} \end{cases}$$

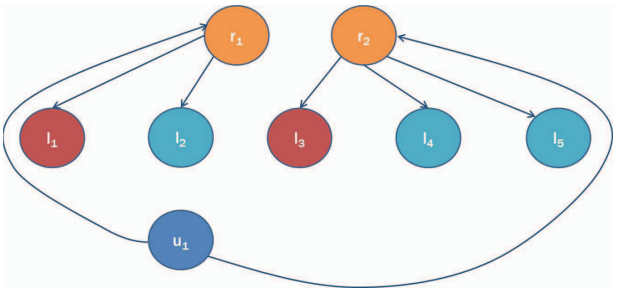


Figure 4: User Region Preferences

For each target user we consider only his regions of interest, disregarding the regions of interest of other users. Figure 4 depicts the graph of Figure 1 with the regions of interest of the target user included. In this example, the target user has equal preferences for regions $r_1$ and $r_2$ with one check-in at each region, although there is only one possible location to be recommended in region $r_1$, which is $l_2$, whereas there are two locations to be recommended in $r_2$. When we apply random walk on this graph, both regions will be visited with the same approximate frequency by the random "walker", although he will have less options within $r_1$ than $r_2$. Thus, $l_2$ will probably be more visited than $l_4$ and $l_5$. For this reason, $l_2$ would be ranked higher than $l_4$ and $l_5$ in this model.

### E. Putting Everything Together

In this subsection we are going to show how to combine the three models presented in the previous subsections into one unified diffusion model. For a given target user $u \in U$, let $G_{\text{unif}} = (V_{\text{unif}}, E_{\text{unif}})$ where $V_{\text{unif}} = V_{cf} \cup V_{\text{dist}} \cup V_{\text{reg}}$, $E_{\text{unif}} = E_{cf} \cup E_{\text{dist}} \cup E_{\text{reg}}$ with edge weights defined as:

$$w^{\text{unif}}(p, q) = \begin{cases} \alpha w^{cf}(p, q), & \text{if } p = u \text{ and } q \in N_u \\ \beta w^{cf}(p, q), & \text{if } p \in N_u \text{ and } q \in L_{N_u} \\ \gamma w^{cf}(p, q), & \text{if } p \in L_{N_u} \text{ and } q \in N_u \\ \delta w^{\text{dist}}(p, q), & \text{if } \{p, q\} \subseteq L_{N_u} \cup L_u \\ \theta w^{reg}(p, q), & \text{if } p = u \text{ and } q \in R \\ w^{\text{reg}}(p, q), & \text{if } p \in R \text{ and } q \in L_p \\ 0, & \text{otherwise} \end{cases}$$

It is important to remark that the values of the hyperparameters $\alpha, \beta, \gamma, \delta$ and $\theta$ cannot be greater than 1. Moreover, since $\alpha$ and $\theta$ weight edges leaving the target user, $\alpha + \theta$ must not be greater than 1. Similarly, $\gamma$ and $\delta$ weigh edges leaving locations, thus $\gamma + \delta$ must not be greater than 1 either.

Differently from the works of [3, 17], we are not generating two or more ranking scores for being combined, but we are rather using one single model that generates one single ranking taking all the assumptions presented in the beginning of this section into account.

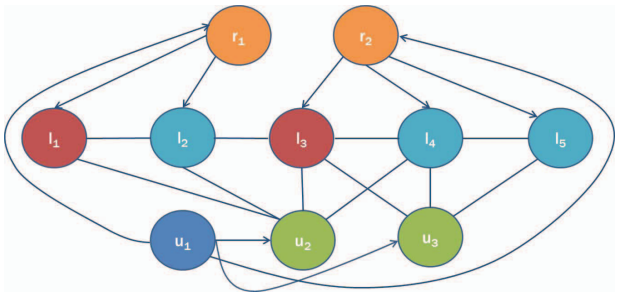


Figure 5: Unified Recommender Model Example

Figure 5 depicts the graph representing our final model. We omitted some edges between the locations is order to improve the readability. For generating recommendations, we perform a random walk on this graph. After the algorithm finishes, we sort the locations in descending order of probability and select the top-5 locations. We will refer to this model as Diffusion Geographic Model (DGM).

## VI. Evaluation

In this section we describe the experimental protocol, data sets, compared recommenders and results achieved. We consider the recommendation scenario where we know exactly the city where the user is currently located. This is a realistic scenario with practical applications since the information of the current city of a user is easy to access in GPS enabled devices, like smartphones and tablets. Moreover, it is more likely that users will visit locations within the city he is currently located than traveling to visit a location in other (sometimes distant) cities.

### A. Data Preparation

We conducted experiments on data from Foursquare[4] and Gowalla[5], two LBSN data sets that were extensively used in the related works [4, 5]. As is usual in the recommender systems literature, we focused on the dense part of the data, i.e., we are not considering the check-ins of the users that have visited less than 10 locations as well as the locations that were visited by less than 10 users.

Before running the recommendation algorithms, we partitioned the original data sets into cities and selected the top-5 cities in number of check-ins. We ended up with three cities from Foursquare (New York, Los Angeles and Chicago) and two from Gowalla (Austin and Stockholm). The characteristics of each city, after filtering the locations and users, are summarized in Table I.

[4]The data was collected by the authors of [4] from September 2010 to January 2011 and is available under request at: http://infolab.tamu.edu/data/

[5]Gowalla was bought by Facebook on 2011 and discontinued on 2012. The data set we used was collected by the authors of [5] from Feb. 2009 to Oct. 2010 and is available at: http://snap.stanford.edu/data/loc-gowalla.html

| City | # Check-ins | $\|U\|$ | $\|L\|$ | Sparsity | LBSN |
|---|---|---|---|---|---|
| New York | 184,760 | 12,005 | 3,073 | 99.49% | Foursquare |
| Austin | 127,625 | 5,385 | 4,000 | 99.40% | Gowalla |
| Stockholm | 90,851 | 5,559 | 3,850 | 99.57% | Gowalla |
| Los Angeles | 64,494 | 6,317 | 1,274 | 99.19% | Foursquare |
| Chicago | 54,600 | 5,268 | 1,090 | 99.04% | Foursquare |

Table I: Data Set Statistics

### B. Compared Algorithms

We compared our approach against the two state-of-art location recommender models described at [3] and [17]. Both approaches use a combination of collaborative filtering, a geographic-aware recommender that models the geographic preferences of users and a social-aware recommender. In this work, we will user only the collaborative filtering and the geographic-aware components of each approach. There are two main reasons that lead us to ignore the social-aware component. The first one is that the information of friendship relations was not collected in the foursquare dataset we used in this work. The second reason is that according to Ye et al. [17] the social relation influence the check-ins mostly when the users travel to distant cities, which is not the scenario we are exploring in this work. We briefly recall these two approaches below.

Based on the idea that there are some regions that the user might be more interested than others, Cheng et al. [3] proposed an approach for combining the preferences of users for regions with his preferences for checked-in locations. For that, first the locations checked-in by users are clustered into regions. Each region is assumed to follow a Gaussian distribution, and the relevance of a new location is computed as the weighted sum of the distances between this location and each of the centroids representing the regions of interest of the target user. These distances are weighted by the importance of a region (number of check-ins at the region) and the probability of the location given a region. For collaborative filtering, it was used probabilistic matrix factorization [15] taking into account the frequency of check-ins of users. It is worth to remark that the combination of these two models was done by a simple multiplication. We will refer to this approach as FMFMGM: Fused Matrix Factorization framework with the Multi-center Gaussian Model. The hyperparameter values of the geographic model of FMFMGM we used in our experiments were $d = 15$, $\alpha = 0.2$, $\theta = 0.02$, as defined in [3]. The hyperparameters values of the matrix factorization model of FMFMGM we used in our experiments were $\alpha = 20$, $\beta = 0.2$, $\lambda = 0.001$ and $k = 10$.

Ye et al. [17] showed that the distances between pairs of checked-in locations of a user follow a power law distribution. After applying a logarithm transformation to this data, the authors propose to learn the parameters of a power law distribution using simple linear regression. The relevance of a new location is then a product over the probabilities

(coming from the fitted power law distribution) of the distances between the new location and all the locations already checked-in at by the target user. For collaborative filtering the authors used K-nearest neighbors with the cosine as similarity function. These two models were fused by a simple linear combination. In our experiments, we set the weight of the geographic recommender to $0.05$ and the weight of the user K-nearest neighbors recommender to $0.95$, cross-validation tuning. Similarly to the original paper will will refer to this model as UG to denote that it combines users ($U$) preferences and with geographic ($G$) influences.

### C. Evaluation Metrics and Protocol

We split the data sets into two distinct sets, the training and the testing sets. For each user in each city data set we randomly removed 10% of his checked-in locations for testing and used the remaining 90% for training. This process was repeated 10 times for each city in order to avoid taking conclusions from biased data. We have a hit for a target user each time the recommendation list of this user contains a test location.

As evaluation metrics we used precision@5 and recall@5. Let $T_u$ be the set of test locations for a given user $u \in U$ and $R_u$ the top-5 recommendation list for this same user. Then precision@5 and recall@5 for a given target user $u \in U$ are defined as follows:

$$\text{precision@5}(u) = \frac{|T_u \cap R_u|}{|R_u|}, \quad \text{recall@5}(u) = \frac{|T_u \cap R_u|}{|T_u|}$$

The hyperparameter values we used in our model are: $k = 80$ (the number of nearest neighbors), $\alpha = 0.5$, $\beta = 0.25$, $\gamma = 0.9$, $\delta = 0.1$, $\theta = 0.25$, $\lambda = 0.1$ (teleport factor) and the restart probability = $0.01$. These values were defined by cross-validation.

### D. Results and Discussion

The results of our experiments are summarized in Figures 6 and 7. These figures depict the average recall@5 and precision@5 for each of the cities described in subsection VI-A averaged over all the 10 random training/testing splits.

Notice that our approach outperforms the compared algorithms in most of the cities evaluated in both precision@5 and recall@5. While DGM and UG achieved similar performance, FMFMGM performed very poorly. One of the reasons for this performance is the extreme level of sparsity of the data sets (cf. Table I), making it very difficult for matrix factorization to learn a reasonable model. The related works have shown that in LBSN the geographical data carries, although important, little signal in comparison to collaborative filtering. Thus, if the collaborative filtering component does not work well, the geographic model alone will perform poorly.

| City | N. York | Austin | Stockholm | Los Angeles | Chicago |
|---|---|---|---|---|---|
| Prec@5 | 0.122 | $1.45E$-05 | 0.358 | 0.0006 | 0.008 |
| Recall@5 | 0.130 | $5.00E$-06 | 0.0305 | 0.02019 | 0.006 |

Table II: Student's paired t test - UG vs DGM

Although the difference is small in comparison to UG in some cities, the difference is statistically significant. For verifying that we conducted a student's paired t-test considering each city and each metric. The null hypothesis of the test was: the UG performance is equal or better than the performance of our approach. Thus the alternative hypothesis is: the UG performance is worse than the performance of our approach. The p-values of the t-tests are summarized in Table II. Since we wanted to achieve 95% of confidence on our conclusions, if the p-value is less than $0.05(1-95\%)$ we can reject the null hypothesis and conclude that the performance of our approach is better than the performance of the UG approach, otherwise we can conclude that either both performances are equal or that the UG performance is better than the performance of our approach.

As we can see from Table II, our approach is better than the UG approach for almost all cities in all metrics, with 95% of confidence. The null hypothesis was not rejected in only 3 out of the 10 tests, i.e., the precision@5 and recall@5 at New York and precision@5 at Stockholm. Since the null hypothesis was accepted for these 3 tests, we know that for each of these scenarios the UG performance might be better or equal to the performance of our approach. Then, for these three scenarios we conducted a second student's paired t-test. Now, the null hypothesis is: the performances of UG and DGM are equal. The p-values of the precision@5 in New York, recall@5 in New York and precision@5 in Stockholm were, respectively, $0.2453$, $0.2604$ and $0.7177$.

Thus, according to these results, we can affirm with 95% of confidence that our approach performs better than UG in check-in data from Los Angeles, Chicago and Austin or presents the same performance in New York. In Stockholm our approach performed better than UG in recall@5 and presented the same performance in precision@5.

## VII. Conclusion

In this work, we introduced a novel diffusion-based location recommendation model, which captures in a single model the users personal taste, the geographic distances between visited locations and regions of interests of the users. This is different from related works where the recommendation models are ensembles of two or three specialized recommenders. We evaluated our model using real world data from two popular location-based social networks: Gowalla and Foursquare. Our experiments showed that our approach outperforms the compared state-of-the-art algorithms in most of the cities evaluated. For future work, we pretend to investigate other contexts of LBSN domain, such as the time of check-ins and the categories of locations.

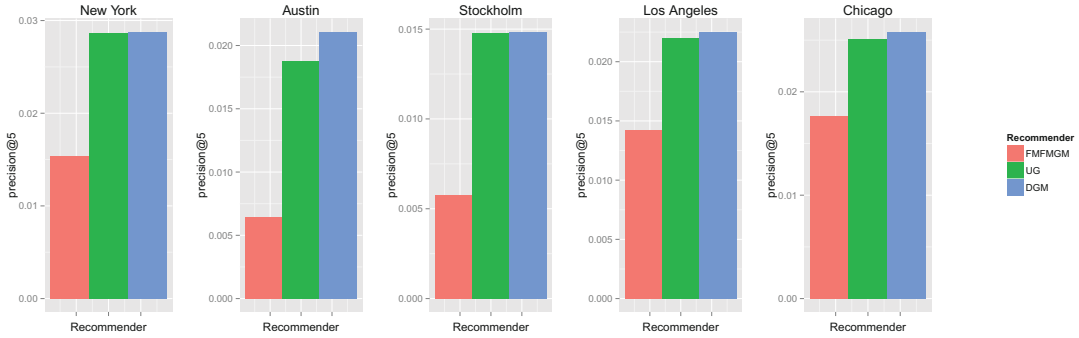


Figure 6: precision@5

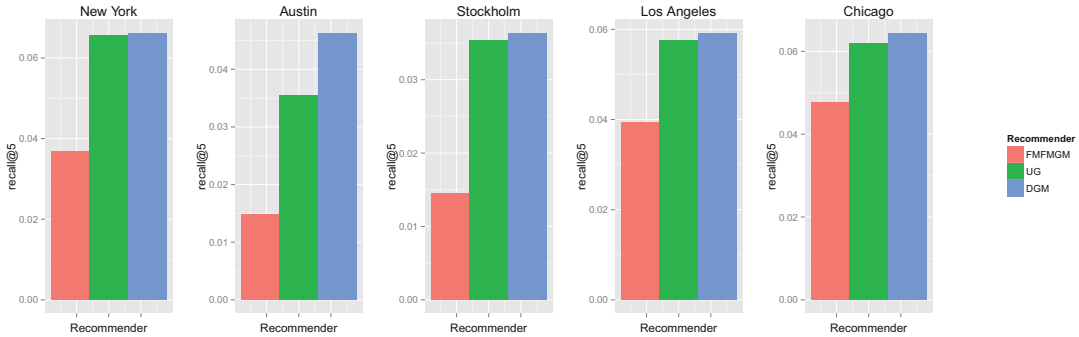


Figure 7: recall@5

## References


[1] Jie Bao 0003, Yu Zheng, and Mohamed F. Mokbel. "Location-based and preference-aware recommendation using sparse geo-social networking data." In: *SIGSPATIAL/GIS*. Ed. by Isabel F. Cruz et al. ACM, 2012, pp. 199–208. ISBN: 978-1-4503-1691-0.

[2] Lars Backstrom and Jure Leskovec. "Supervised Random Walks: Predicting and Recommending Links in Social Networks". In: *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*. WSDM '11. Hong Kong, China: ACM, 2011, pp. 635–644. ISBN: 978-1-4503-0493-1.

[3] Chen Cheng et al. "Fused Matrix Factorization with Geographical and Social Influence in Location-Based Social Networks". In: *AAAI*. 2012.

[4] Zhiyuan Cheng et al. "Exploring Millions of Footprints in Location Sharing Services." In: *ICWSM*. Ed. by Lada A. Adamic, Ricardo A. Baeza-Yates, and Scott Counts. The AAAI Press, 2011.

[5] Eunjoon Cho, Seth A Myers, and Jure Leskovec. "Friendship and mobility: user movement in location-based social networks". In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2011, pp. 1082–1090.

[6] Huiji Gao et al. "Exploring temporal effects for location recommendation on location-based social networks." In: *RecSys*. Ed. by Qiang Yang 0001 et al. ACM, 2013, pp. 93–100. ISBN: 978-1-4503-2409-0.

[7] Bo Hu and Martin Ester. "Spatial topic modeling in online social media for location recommendation." In: *RecSys*. Ed. by Qiang Yang 0001 et al. ACM, 2013, pp. 25–32. ISBN: 978-1-4503-2409-0.

[8] Robert Jäschke et al. "Tag Recommendations in Social Bookmarking Systems". In: *AI Commun.* 21.4 (Dec. 2008), pp. 231–247. ISSN: 0921-7126.

[9] Ioannis Konstas, Vassilios Stathopoulos, and Joemon M. Jose. "On Social Networks and Collaborative Recommendation". In: *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '09. Boston, MA, USA: ACM, 2009, pp. 195–202. ISBN: 978-1-60558-483-6.

[10] Takeshi Kurashima et al. "Geo Topic Model: Joint Modeling of User's Activity Area and Interests for Location Recommendation". In: *Proceedings of the Sixth ACM International Conference on Web Search*

*and Data Mining*. WSDM '13. Rome, Italy: ACM, 2013, pp. 375–384. ISBN: 978-1-4503-1869-3.

[11] Xin Liu et al. "Personalized point-of-interest recommendation by mining users' preference transition." In: *CIKM*. Ed. by Qi He et al. ACM, 2013, pp. 733–738. ISBN: 978-1-4503-2263-8.

[12] Anastasios Noulas et al. "An Empirical Study of Geographic User Activity Patterns in Foursquare." In: *ICWSM*. Ed. by Lada A. Adamic, Ricardo A. Baeza-Yates, and Scott Counts. The AAAI Press, 2011.

[13] Iury Nunes and Leandro Marinho. "A Gaussian Kernel Approach for Location Recommendations". In: *Proceedings of KDMiLe - Symposium on Knowledge Discovery, Mining and Learning, ISSN 2318-1060*. 2013.

[14] L. Page et al. "The PageRank citation ranking: Bringing order to the Web". In: *Proceedings of the 7th International World Wide Web Conference*. Brisbane, Australia, 1998, pp. 161–172.

[15] Ruslan Salakhutdinov and Andriy Mnih. "Probabilistic Matrix Factorization". In: *Advances in Neural Information Processing Systems*. Vol. 20. 2008.

[16] Thiago H. Silva et al. "You are What you Eat (and Drink): Identifying Cultural Boundaries by Analyzing Food & Drink Habits in Foursquare". In: *CoRR* abs/1404.1009 (2014).

[17] Mao Ye et al. "Exploiting geographical influence for collaborative point-of-interest recommendation." In: *SIGIR*. Ed. by Wei-Ying Ma et al. ACM, 2011, pp. 325–334. ISBN: 978-1-4503-0757-4.

[18] Hongzhi Yin et al. "LCARS: a location-content-aware recommender system." In: *KDD*. Ed. by Inderjit S. Dhillon et al. ACM, 2013, pp. 221–229. ISBN: 978-1-4503-2174-7.