Location Recommendation Based on Periodicity of Human Activities and Location Categories

Seyyed Mohammadreza Rahimi and Xin Wang

Department of Geomatics Engineering, Schulich School of Engineeting,
University of Calgary, 2500 University Dr. N.W. Calgary, Alberta, Canada, T2N 1N4
{smrahimi,xcwang}@ucalgary.ca

Abstract. Location recommendation is a popular service for location-based social networks. This service suggests unvisited sites to the users based on their visiting history and site information. In this paper, we first present how to build the temporal and spatial probability distribution functions (PDF) to model the temporal and spatial checkin behavior of the users. Then we propose two recommender algorithms, Probabilistic Category Recommender (PCR) and Probabilistic Category-based Location Recommender (PCLR), based on the periodicity of user checkin behavior. PCR uses the temporal PDF to model the periodicity of users' checkin behavior. PCLR combines the temporal category model used in PCR with a geographical influence model built on the spatial PDF. The experimental results show that the proposed methods achieve better precision and recall than two well-known location recommendation methods.

Keywords: Recommender system, Location-based Social Networks, Location-Category, probability model.

1 Introduction

In location-based social networks (LBSN), people share location-related information with each other, and also leverage collaborative knowledge learned from user-generated and location-related content. Among various LBSN services, the location recommendation service suggests unvisited sites to the users based on the information collected on LBSNs, such as checkins, social ties, user profiles and location profiles.

The location recommendation has been an active research area. The existing methods focus on the "geographical influence" and the "social influence" on users checkin behavior [1, 2, 4]. Modeling the geographical influence, the recommender finds the probability of a user visiting locations based on the distance of the locations to the user's home [2] or to the previously visited locations by that user [1]. The challenge of the existing geographical influence-based methods is that they do not consider temporal effect on human checkin behaviors. Given a user's checkin history, the methods would recommend the same set of locations regardless of noon or midnight. The other attempt in the literature is to make use of the social ties among users. The assumption is that people have the similar checkin patterns with their friends. However, only 10~30% of the checkins are influenced by social links [2].

J. Pei et al. (Eds.): PAKDD 2013, Part II, LNAI 7819, pp. 377–389, 2013.

In this paper, we approach the problem from an activity-based perspective. We believe that people behave on a periodic pattern. In the other words, people are more likely to conduct the same activity around the similar time of the day. Such temporal pattern exists but it is different from one person to another. Since the category of a location reflects the activities happening in that location, we believe a similar temporal pattern exists for the location categories. After analyzing checkin data of a real LBSN, we find such a pattern for the categories of locations. We also find that the further a location is away from the home location of the user, the lower chance he or she will visit that location. The probability decreases exponentially as the distance increases. But the degree of the decrease varies for different reaching distances. Thus, we could make better location recommendation if we divide the reaching distance of the user into home and away zones and find the spatial probability distribution functions for each zone separately.

In this paper, we combine the periodicity of user's behavior and the geographical influence into location recommendations. Specifically, our contributions in this study include:

- We present how to build the temporal and spatial probability distribution functions (PDF) to model the temporal and spatial checkin behavior of the users. The temporal PDF models the periodic pattern of user checkins. It is discovered in temporal analysis of the checkins. The spatial PDF is discovered in the spatial analysis of the checkins and models the probability of checking in to a location as a function of the distance of that location to user's home.
- We propose a category recommender algorithm called Probabilistic Category Recommender (PCR). It recommends category of locations to the users at a given time using the temporal PDF and the checkin history of those users.
- Based on PCR, we propose a location recommender algorithm called Probabilistic Category-based Location Recommender (PCLR) which uses the PCR's category model along with a spatial model. The spatial model is built upon the spatial PDF and measures the probability of checking in to a location based on the distance of that location to user's home. PCLR combines the probabilities of PCR's category model and the spatial model to find the probability of checking in to a location.
- We conduct the experiments on a real LBSN checkin dataset to evaluate the performance of the PCR and PCLR algorithms. We discover that the category recommendation (PCR) is more effective compared to the exact location recommendation (PCLR). In addition, PCLR outperforms two existing well-known location recommenders, PMM and USG.

The paper is organized as follows: in Section 2, we give a literature review on location recommendation algorithms for LBSNs. Section 3 introduces the dataset used in this paper and confirms the assumptions used in the algorithms through the data analysis. In Sections 4 and 5, we propose PCR and PCLR algorithms, respectively. The experimental results are presented in Section 6. Finally, we conclude the study in Section 7.

2 Related Works

The location recommendation is an active research area. Zheng et al. [5] recommend locations to the users based on the real-world location history of the users collected in GPS trajectories. Park et al. [6] proposed a method of providing personalized location recommendation to the users based on the location history of the users. Simon et al. [7] and Beeharee et al. [8] used the mobile tour guide systems which collect real-time location of the users for location recommendations. Zhou et al. [4] proposed a method called probabilistic latent semantic analysis (PLSA). It first trains a latent semantic model, and then it uses that model to find the probability of a user checking in to a given location. However, none of these methods consider the fact that human checkin behavior is influenced by the distance of user to the location of checkin. To add on, they do not use the temporal patterns of human checkin behavior for location recommendation.

Ye et al. [1] proposed a fusion framework USG consisting of three different models 1) a user-based collaborative filtering (CF) model, 2) a social influence model and 3) a geographical influence model. The user-based CF model estimates the implicit preference of a user for a location combining the behavior of similar users. The social influence model, which also is a CF model, estimates the implicit preference of a user by aggregating the behavior of his or her friends. Finally, the geographical influence model uses a power law distribution to find the probability of checkin at given distances from the users previously visited locations. The method builds different models for different aspects of location recommendation and provides a fusion framework for combining these models. However, applying the CF method for location recommendation may not be suitable since the similar users in a location recommendation approach might be in different locations. Therefore, recommending one's behavior to the other is not appropriate. Another trend in the recent research is making preference-aware recommendations using the location categories. Bao et al. [10] make recommendations using the checkin history of the local experts. Local experts are users with high expertise in user's preferred categories and the venues in the geospatial range of the user. However, their method lacks the temporal feature for recommendation which can be modeled using a periodic movement model for the users.

In the more recent researches the periodicity of the human behavior has gained the attention of researchers [2, 9, 11, 12, 13]. Eagle et al, [13] model the behavior of an individual using the weighted sum of a set of characteristic vectors called "eigenbehaviors". Li et al. [10, 11] define the periodic behavior of a moving object as "the repeating activities at certain locations with regular time intervals." and then mine the periodic movements of moving objects. Cho et al. [2] propose a location recommendation method based on the periodicity of the human movement. They propose two methods PMM (Periodic Mobility Model) and PSMM (Periodic Social Mobility Model). In PMM the user can be in home or work states and being in different states is defined using a temporal probability distribution function. The PSMM is based on PMM and adds the effect of social ties to it. However, the two methods assign the training checkins to different states randomly and iterate until an optimal classification is reached. Therefore, the checkins classified as work or home state might be belonging to the other group.

3 Dataset Description and Data Analysis

In this section, we will first describe the Gowalla data used in the paper. Then we will confirm two assumptions used by our location recommendation algorithms with the Gowalla data and illustrate how to build the probability distribution functions based on the data analysis.

The dataset used in this paper is collected from Gowalla, which was one of the popular online LBSN services until it was closed in 2012 (for details of the data crawler and data collection see [4].) The dataset contains 5462 users, 5999 locations and 104851 checkins. A checkin indicates a user has visited a location at certain time. In our dataset, a checkin includes user-id, spot-id, spot-latitude, spot-longitude, spot-category and timestamp. Spot-latitude and spot-longitude are the latitude and longitude of the checked-in location. Spot-category is the category of the checkin location, for example, "Coffee shop" or "Office". Finally, the timestamp of the checkin shows the date and time the user visited the spot.

Before introducing our algorithms, we will confirm two underlying assumptions of our algorithms on Gowalla dataset. These two assumptions are: 1) People have temporal patterns for their daily activities and checkins (confirmed in temporal analysis); 2) People visit locations closer to their home with a higher probability compared to the further locations (confirmed in spatial analysis).

3.1 Temporal Analysis

We believe that people have a periodic behavior for visiting similar type of locations. For example, a person might go to coffee shops everyday at 8am but she might go to different coffee shops on different days. To test this, we first find the pairs of checkins to the same category of location from the same user, and then we plot the frequency of checkin pairs based on the time interval of those checkins. Fig. 1 shows the plot of frequency of checkin to the same category at given time differences using 1 hour time window.

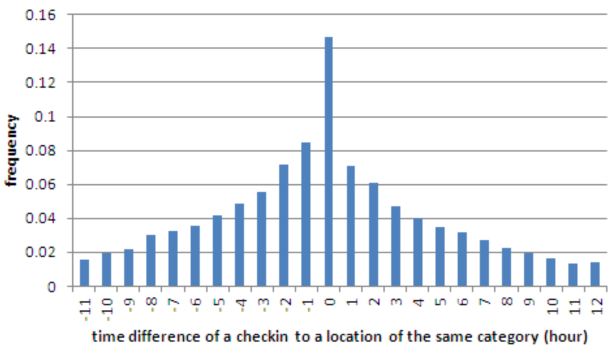

Fig. 1. Frequency of checkins to locations of the same category to the time difference of those checkins using 1 hour time window

As shown in Fig. 1, using the one-hour time window the probability of checkin is the highest (about 15%) for 0 hour time difference and it declines as the absolute value of the time difference increases. Adding up the frequency of checkins from -2 to 2 hour time difference, we find out that the probability of checking in to the locations from the same category is about 45% for the mentioned 5-hour time window. As an example, if a user checks in to a coffee shop at 8 am, the chance that she will checkin to a coffee shop (the same coffee shop or a different one) between 7:30am to 8:30am in the coming days is 15%, and the probability that she will checkin to a coffee shop between 5:30am and 10:30am in the coming days is about 45%.

After plotting the frequency of checkin to the same category given the time difference, we can make a temporal Probability Density Function (PDF) based on the plot. This PDF helps us quantify the probability of checkin to different categories at different times of the day. To do so, we first define the function *F*, consisting of 24 different constant outputs and the values of the outputs based on the checkin plot. For example the *F(Δt)* for Fig. 1 is defined as:

$$F(\Delta t) = \begin{cases} .15, & \lfloor \Delta t \rfloor = 0 \\ .07, & \lfloor \Delta t \rfloor = 1 \\ .085, & \lfloor \Delta t \rfloor = -1, \\ & \quad \ldots \\ .0018, & \lfloor \Delta t \rfloor = 12 \end{cases} \tag{1}$$

where Δt is the time difference and $\lfloor \Delta t \rfloor$ indicates the floor of t. Based on this, we can define the temporal probability distribution function TP for a given set of checkins as:

$$TP(t;\mu) = F(\lfloor t-\mu \rfloor), \tag{2}$$

where t is the time we compute the probability for and μ is the average time of the checkins in the subset.

3.2 Spatial Analysis

We believe the probability of a user visiting a location closer to their home location is higher than the probability of visiting locations farther from their homes. However, the home locations of the users are usually not given in the dataset. To find the home location, we assume that user checkins are centered at his home location. We are finding the home location of a user by averaging the locations visited by the user. But this estimation could be affected by some checkin locations when the user was on a trip, especially for the users with small number of checkins. To solve this problem, we first divide the surface of the earth into small non-overlapping regions and then find the region with the most number of checkins [2]. We consider the average point of the locations in that region as the candidate home location of the user. However, because we use fixed regions and average the locations in each of those regions, there's a chance that we are missing the actual home location. To solve this, we select all the checkins by the user in 100km radius of the candidate home location. 100 km is used as the human reach distance based on [1]. Finally, we average all the locations in this selected set of checkins to find the home location of the user.

Algorithm 1 shows the steps of finding the home location of the user using the user's checkin history. It first groups the checkins based on their regions (lines 1 to 3). In line 4 we select the region with maximum number of checkins, and find the average location of the checkins in the region as the candidate home location (line 5). Then we find all of the checkins in the reaching distance of the candidate home location (lines 6 to 10) and return the average of those locations as the home location of the user (line 11).

After having the home location of users, we will test whether the frequency of checking in to locations decreases as the distance to a user's home location increases. To do so, we first calculate the distance between locations checked in by the users and their home location and then compute the frequency of checking in at any distance of their home location. Fig. 2 shows the logarithmic scale plot of the probability of checkin over the distance to the home location of the user.

```
Algorithm 1 findHomeLocation (checkins)
// checkins is a set of checkins of a user
Begin
01-for each (c in checkins)
02-    region[c].add(c);
03-end for
04-selectedRegion <- maximumSized(region);
05-candidateHomeLocation <- average(checkins In selectedRegion);
06-for each (c in checkins)
07-    if (c is in reachingDistance of candidateHomeLocation)
08-      selectedCheckins.add(c);
09-    end if
10-end for
11-return average(selectedCheckins);
End.
```

Fig. 2 shows that: first, the checkin frequency values for distances greater than 50km vary randomly (shown as triangular points), which means that 50km is the range of human checkin behavior for our dataset and checkins happen on distances greater than 50km when the user is on a trip. Second, based on the slope of the linear relationship, we can separate the less than 50km part into two different parts, less than 16km (shown as diamond shaped points) and greater than 16km (shown as rectangular points). We can tell that the probability decreases more slowly in less than 16km part. In this case, we assume that the area within 16km radius of the user's home location is his home zone and the outside area is the away zone. Based on these findings we should use different PDFs for each of these zones in order to have a better fitting model.

To find the spatial PDF, we use exponential estimation to find the relationship of frequency and distance in each of the mentioned zones. Based on Fig. 2 we define the spatial probability distribution (SP) for Gowalla dataset as:

$$SP(l;h) = \begin{cases} 0.0886e^{-0.166*distance(l,h)}, & distance(l,h) \leq 16Km \\ 0.3122e^{-0.204*distance(l,h)}, & 16Km < distance(l,h) \leq 50Km, \\ 0, & 50Km < distance(l,h) \end{cases} \quad (3)$$

where l is the location for which we want to find the probability of checkin and h is the home location of the user.

4 Probabilistic Category Recommender

As discussed in Section 3.1, users are more likely to checkin to the locations of the same category around the same time of the day. Based on this, the user's checkin behavior and their checkin location categories can be used to predict the category of the location the user is going to visit. Additionally, this can be used to assign categories to uncategorized locations in dataset based on the behavior of the user explored those locations.

In this section, we describe how we build a temporal model based on the user behavior and how it is used for recommending the categories of locations. The temporal model is a user-specific model. Each user has a different model that is trained based on their checkins. For a given user and a given time of the day, this model will return a list of categories and the probability values that user will visit a location belonging to each of those categories.

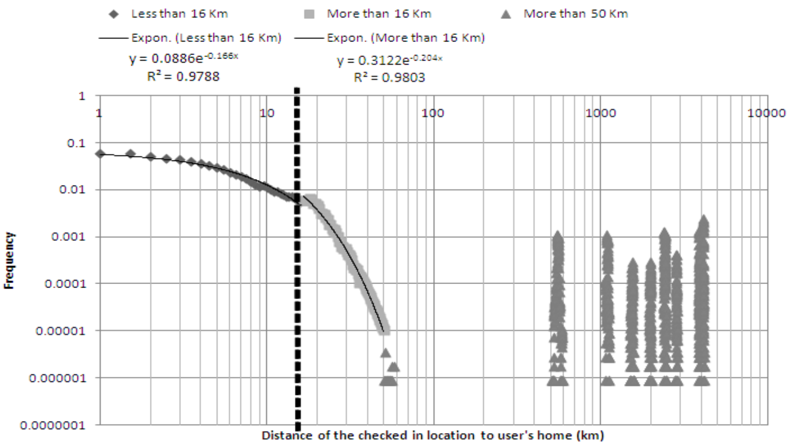

Fig. 2. Logarithmic scale of the frequency of checkins to the distance to user's home

To make this model, we start with the checkin history of the users. Then in order to find the similar checkins we separate the checkins into subsets based on their category and time. Doing this, we find checkin subsets where each subset contains checkins which have happened to locations of the same category and the timestamps showing the same time window. For example, all checkins to the coffee shops that happened between 4:00pm-4:59pm are put in one subset and checkins to coffee shops happened between 8:00am-8:59am are put in another subset. In order to build the temporal PDF introduced in the previous section, we need to find the average time of the checkins in each subset of checkins. Because we have one temporal PDF for each of the checkin subsets, we need a weighting value to normalize these temporal PDFs so that the whole model satisfies the second axiom of probability.

The average time of each subset helps us find the central point of the checkins of that specific type. The average time of the subset s_i is calculated as $\mu_{s_i} = \frac{\sum_{c_j \in s_i} t_{c_j}}{|s_i|}$, where s_i is a subset of checkins by a user. c_j is a checkin selected from s_i. μ_{s_i} is the average time of the checkins in s_i and t_{c_j} is the time of checkin c_j. The weight of each subset is the number of checkins in that subset divided by the whole number of checkins by that user. The weight of the subset s_i is defined as $w_{s_i} = \frac{|s_i|}{|checkins\ by\ u|}$, where $|s_i|$ the number of checkins in s_i. A larger value of weight shows that more checkins are assigned to that subset and hence that subset is of more importance.

Next, we calculate the probability of checkin to the category of that subset based on the temporal PDF.

The probability of user *u* checking in a location of category *c* at time *t* is:

$$T(u, c\,|t) = \sum_{\{s_i \in subsets[u] | category[s_i] = c\}} w_{s_i} * TP(t; \mu_{s_i}). \tag{4}$$

This equation shows how to compute the temporal probability of a given category. The probability of checking in to a category is the summation of the weight of all subsets by that user matching the given category multiplied by the temporal probability of that subset at the given time. In this equation s_i is i-th subset of checkins with category *c* and w_{s_i} is the weight of subset s_i. *TP* is the temporal probability distribution function introduced in Eq. 2. Finally, μ_{s_i} is the average time of the checkins in subset s_i.

```
Algorithm 2 buildCategoryModel (checkins, u)
// checkins is a set of checkins of an user u
Begin
For each (c in checkins)
    category <- getLocationCategory(c);
    h <- getHourOfDay(c.time);
    addToSubset(u.subsets[category,h],c);
endfor;
For each (s in u.subsets)
    Weight[s] <- size(s)/size(checkins);
    Average[s] <- averageTimeOfCheckins(s);
End for;
End.
```

```
Algorithm 3 calculateCategoryProbability (u, category, time)
Begin
P <- 0;
For each (s in u.subsets)
    If (category (s) = category )
        P <- P + weight[s] * F ([time - average[s]]);
    End if
End For
Return P;
End.
```

```
Algorithm 4 PCRrecommendCategories (u, time, k)
Begin
for each (c in categories)
    probability[c] <- calculateCategoryProbability(u , c , time);
end for
sortedCategories <- sort(categories based on probability);
return sortedCategories[1] ... [k];
End.
```

The algorithms for building the model and finding the probability of checkins are given in Algorithm 2, 3 and 4. The `buildCategoryModel` algorithm (Algorithm 2) is responsible for making the temporal probability model and it is the starting point of the PCR method. As stated earlier, the first step in building the PCR model is to group the checkins into subsets of checkins with the same category and the same time window. So the first loop (lines 1 to 5) separates the checkins into subsets based on the category and the time. The next step is to find the weight of the subsets and average time of the checkins in each subset. The second loop (lines 6 to 9) is responsible for this task.

In order to find the probability of each category at given times, we need to calculate the temporal probability introduced in Eq. 4. The Algorithm 3 `calculateCategoryProbability` finds the probability of checkins to a specific category at a given time for an individual user. The algorithm needs to first find the subsets belonging to the given category for the user. The loop starting on line 2 loops over all subsets, and using an If-statement finds the satisfying subsets. In line 4, the probability is computed using the Eq. 4.

Now we have the methods for building the model and calculating the probability of each category we can make the recommender algorithm (Algorithm 4). To make the category recommendation, we loop over all categories and calculate the probability of checkin to that category at the given time (lines 1 to 3). Then we sort the categories based on the probability (line 4) and return the top-k categories to the users (line 5).

Example: Consider a user with checkins to 8 locations. 3 of them to coffee shops around 8:15, 2 of them to coffee shops around 17:20 and the remaining three to fast food restaurants around 17:20. This user requests for recommendation at 18:01 on Sept 21st 2012. To recommend a category based on his checkin history, the first step is to build the PCR model using the Algorithm 2. So first we separate his checkin history into three different subsets (s1:"Coffee shop, around 8:15",s2:Fast food around 12:30. and s3:Coffee shop, around 17:20). Then we find the weight and average time of each subset: w_1 = 3/8, w_2=3/8, w_3 = 2/8, μ_1 = 8:15, μ_2 = 12:30 and μ_3 = 17:20. Now we have the PCR temporal model. The next step is finding the probability of different categories for the given time. Following the steps of Algorithm 3 the probability of checkin to a coffee shop and a fast food can be calculated as follows:

$$P(John, coffee\ shop\ |t) = w_1 * TP(time; \mu_1) + w_3 * TP(time; \mu_3)$$
$$= .375 * F\ (\lfloor 18{:}01 - 8{:}1\rfloor + .25 * F(\lfloor 18{:}01 - 17{:}20\rfloor) = 0.0425$$
$$P(John, fast\ food\ |t) = w_2 * TP(time; \mu_2) = .375 * F\ (\lfloor 18{:}01 - 12{:}30\rfloor) = 0.013$$

5 Probabilistic Category-Based Location Recommender

In this section, we propose a new location recommendation algorithm called Probabilistic Category-based Location Recommender (PCLR). PCLR extends from PCR and it combines the geographical influence and the recurring pattern of the user activities to improve the location recommendation. The idea of PCLR is to first find the locations the user is most likely to visit based on the geographical influence and then, in order include the effect of time, weight those locations using the probability values of the category of those locations which is done using the PCR algorithm.

The spatial component of the PCLR algorithm which is responsible for the geographical influence is based on the probability distribution suggested in Eq. 3; it also uses the PCR algorithm to weight the recommended locations found using their geographical influence. Combining these two, the probability of user (u) checkin to the location (l) at the given time (t) is defined as:

$$P(u, l\,|t) = SP(l; home_u) * \mathrm{T}(u, c_l|t), \tag{5}$$

where $home_u$ is the home location of user u and c_l is the category of location l. SP is the spatial probability of visiting location l given the home location of the user (Eq. 3) and T is the temporal probability of checking in the category of location l at given time t (Eq. 4).

For the previous example, if a coffee shop is in the distance of 5km from user's home location. The spatial probability that the user checks in that location is 0.024. The temporal probability of checkin to a coffee shop as we computed in the previous section is 0.0425. Then the probability of the sample user checking into that specific coffee shop is: $sp * tp = 0.024 * 0.0425 = 0.00102$

The PCLR algorithms are shown in Algorithms 5, 6 and 7. The `buildPCLRModel` algorithm (Algorithm 5) is responsible for making the PCLR model. It first makes the PCR model in line 1, and then finds the home location of the user using Algorithm 1 (line 2).

Algorithm 6 `calculateLocationProbability` computes the probability of a checkin for an individual user to a certain location at a given time using Eq. 5. The first line of this algorithm finds the spatial probability of location using the Eq. 3. The second line calls the `calculateCategoryProbability` to find the probability the user will checkin to a location of the same category as location l. Line 3 finds the probability of checkin to the location by multiplying these two probabilities as is suggested earlier in Eq. 5.

The PCLR location recommender (Algorithm 7). first finds the probability of the user checking in to candidate locations (lines 1 to 3) and recommends the top-k locations to the user (lines 4 and 5).

```
Algorithm 5 buildPCLRModel ( checkins , u)
// checkins is a set of checkins of user u
Begin
1- buildCategoryModel( chechins, u)
2- home[u] <- findHomeLocation( checkins )
End.
```

```
Algorithm 6 calculateLocationProbability (u, l, t)
// Returns the probability user u check in to location l at time t
Begin
// SP is the PDF introduced in the section 3.2.
1- spatialProbability <- sp(l;h);
2- tp <- calculateCategoryProbability (u, category[l], t);
3- return spatialProbability * tp;
End.
```

```
Algorithm 7 PCLRrecommendLocations (u, time, k)
Begin
1- for each (l in locations)
2-   probability[c] <- calculateLocationProbability (u , l , time);
3- end for
4- sortedLocations <- sort(locations based on probability);
5- return sortedLocations[1] ... [k];
End.
```

6 Experiments

We implement the algorithms in Java and use a Mac with 8GB of ram and a 2.3GHz, Intel Core i5 CPU for the experiments. We divide the Gowalla data described in Section 3 into the training and testing datasets. To do so, we randomly move one of the checkins of every user to the testing dataset and leave the rest in the training dataset. As the result, the training dataset contains 99389 checkins and the testing dataset contains 5462 checkins. We randomly generated 5 groups of different training and testing datasets and report the average performance from five runs.

To evaluate the performance of the algorithm, we use Precision and Recall. Precision is the ratio of the number of relevant instances to the number of retrieved instances, while recall is the ratio of the number of relevant instances retrieved to the number of relevant instances. They are defined as:

$$Precision = \frac{Number\ of\ correct\ recommendations}{Number\ of\ recommendation}$$

$$Recall = \frac{Number\ of\ correct\ recommendations}{Number\ of\ correct\ answers}$$

We compare the performance of our proposed algorithms with two existing methods, Periodic Mobility Model (PMM) proposed by Cho et al. [2] and the USG model proposed by Ye et al [1]. However, because these two methods do not provide a category recommender, we modify them in order to have a fair comparison with the PCR algorithm. We change the two methods to return the categories of the top locations instead of the exact location itself. We call the new PMM and USG methods PMM+c and USG+c, respectively. Fig. 3 shows the precision and recall values of different recommender algorithms.

From Fig. 3, we can tell that both PCR and PCLR perform better than competing methods. Considering the exact location recommender algorithms, we find out that PCLR outperforms both PMM and USG regardless precision and recall (Fig.3. (a) and (b)). This proves that using the location categories and periodic user behavior help improve the location recommendation. We also discover that PMM outperforms USG. This could be the result of PMM benefited from a periodic model of human movements which USG did not.

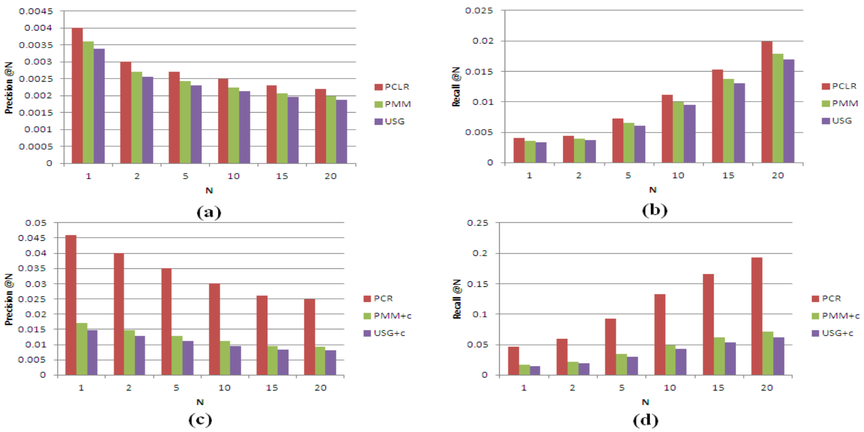

Fig. 3. Comparison of different recommender algorithms on Gowalla checkins. (a) Precision of PCLR, PMM and USG location Recommenders. (b) Recall of PCLR, PMM and USG location Recommenders. (c) Precision of category recommender algorithms. (d) Recall of category recommender algorithms.

As for the category recommender algorithms, PCR performs better than PMM+c and USG+c with a big margin considering both precision and recall (Fig. 3 (c) and (d)). The reason is that PCR is specifically built for category recommendation. It uses the temporal PDFs derived from the dataset for the categories. We also observe that PMM+c performs better than USG+c. Again we think this is because PMM+c uses a periodic model.

7 Conclusions and Future Work

In this paper, using the data collected from Gowalla, we discover that users have a recurring behavior of visiting locations over time. We also find that users are more likely to visit locations near their home. Based on these findings, two recommenders, a category recommender (PCR) and a location recommender (PCLR), are proposed. PCR provides suggestions with the location category for the next user visit. PCLR provides recommended locations to the users at a given time of the day. Experimental results show that the recommending location category to the user is more effective than the exact location. Our proposed algorithms perform much better than the existing well-known methods.

In the future, we will build more complicated models for both spatial and temporal components. We will evaluate our algorithms on larger datasets and compare it with other existing models as well. To add on, we are also planning to study the relationship of the social ties and the user checkin behaviors in order to improve our current models.

Acknowledgement. The research is supported by Canada NSERC Discovery Grant.

References

1. Ye, M., Ying, P., Lee, W., Lee, D.: Exploiting Geographical Influence for Collaborative Point-of-Interest Recommendation. In: 34th ACM International Conference on Research and Development on Information Retrieval, Beijing, China, pp. 325–344 (2011)
2. Cho, E., Myers, S., Leskovec, J.: Friendship and Mobility: User Movement In Location-Based Social Networks. In: 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, San Diego, California, USA, pp. 1082–1090 (2011)
3. Cheng, Z., Caverlee, J., Lee, K., Sui, D.: Exploring millions of footprints in location sharing services. In: 5th International Conference on Weblogs and Social Media, Barcelona, Spain, pp. 81–88 (2011)
4. Zhou, D., Wang, B., Rahimi, S.M., Wang, X.: A Study of Recommending Locations on Location-Based Social Network by Collaborative Filtering. In: Kosseim, L., Inkpen, D. (eds.) Canadian AI 2012. LNCS, vol. 7310, pp. 255–266. Springer, Heidelberg (2012)
5. Zheng, V.W., Zheng, Y., Xie, X., Yang, Q.: Collaborative Location and Activity Recommendations with GPS History Data. In: 19th International Conference on World Wide Web, Raleigh, North Carolina, USA, pp. 1029–1038 (2010)
6. Park, M.-H., Hong, J.-H., Cho, S.-B.: Location-Based Recommendation System Using Bayesian User's Preference Model in Mobile Devices. In: Indulska, J., Ma, J., Yang, L.T., Ungerer, T., Cao, J. (eds.) UIC 2007. LNCS, vol. 4611, pp. 1130–1139. Springer, Heidelberg (2007)
7. Simon, R., Frőhlich, P.: A Mobile Application Framework for the Geospatial Web. In: 16th International Conference on World Wide Web, Banff, Alberta, Canada, pp. 381–390 (2007)
8. Beeharee, A., Steed, A.: Exploiting Real World Knowledge in Ubiquitous Applications. Personal and Ubiquitous Computing Archive 11(6), 429–437 (2007)
9. Wang, J., Prabhala, B.: Periodicity Based Next Place Prediction. In: Workshop on Mobile Data Challenge by Nokia, Newcastle, UK (2012)
10. Bao, J., Zheng, Y., Mokbel, M.: Location-based and Preference-Aware Recommendation Using Sparse Geo-Social Networking Data. In: 20th ACM SIGSPATIAL International Conference on Advances in GIS. Redondo Beach, California (2012)
11. Li, Z., Ding, B., Han, J., Kays, R., Nye, P.: Mining periodic behaviors for moving objects. In: 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, pp. 1099–1108 (2010)
12. Li, Z., Wang, J., Han, J.: Mining event periodicity from incomplete observations. In: 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Beijing, China, pp. 444–452 (2012)
13. Eagle, N., Pentland, A.: Eigenbehaviors: identifying structure in routine. Behavioral Ecology and Sociobiology 63, 1057–1066 (2009)