Location, Time and Preference Aware Restaurant Recommendation Method

by

Md. Ahsan Habib and Md. Abdur Rakib

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE
COMPUTER SCIENCE AND ENGINEERING
BRAC UNIVERSITY

AUGUST, 2016

Declaration

This is to certify that the research work titled "Location, Time and Preference Aware Restaurant Recommendation Method" is submitted by Md. Ahsan Habib and Md. Abdur Rakib to the Department of Computer Science and Engineering, BRAC University in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering. We hereby declare that this thesis is based on results obtained from our own work. The materials of work found by other researchers and sources are properly acknowledged and mentioned by reference. This thesis, neither in whole nor in part, has been previously submitted to any other University or Institute for the award of any degree or diploma. We carried our research under the supervision of Dr. Muhammad Abul Hasan.

Supervisor:

Dr. Muhammad Abul Hasan
Assistant Professor
Department of Computer Science and Engineering
BRAC University, Dhaka

Authors:

Md. Ahsan Habib	Md. Abdur Rakib
12101077	12101036

BRAC
UNIVERSITY

Inspiring Excellence

Dedication to our parents.

Acknowledgments

We would like to start by thanking our thesis supervisor Dr. Muhammad Abul Hasan for allowing us to work on this thesis under his supervision and for his inspiration, ideas and suggestions to improve this work. He has offered us help to understand the necessary concepts and supported at many difficult stages of our work. We are thankful to our faculty members, specially to Annajiat Alim and Matin Saad Abdullah for helping to walk into the path of Computer Science. Through their continuous assistance and inspiration, we have been able to utilize and enrich our knowledge in this field. We are also thankful to Professor Andrew for providing great open resources about Machine Learning which helps us to learn and understand many sophisticated concepts on machine learning. Finally, we are great full to our parents for their continuous support. We also would like to thank all of our friends and family members who help us directly or indirectly in this journey.

Abstract

Location based social networks (LBSN) introduce a platform to understand users' preference via analyzing their check-in history. Such data are being used in the literature for wide variety of location aware recommendation systems. In this thesis, we propose a novel location, time and preference aware restaurant recommendation method by using users' check-in history, user's current geospatial location and current time. In the proposed method, each user's check-in history is modeled individually to discover the preference trend by using a logistic function. At the same time, each restaurant's popularity is calculated using user-restaurant mutual reinforcement learning. The restaurant recommendation scores are computed by considering four key factors, namely, i) user's preference score ii) the distance of a venue; iii) the time of a day; and iv) popularity of a venue. Each of these key factors are modeled carefully to estimate a realistic recommendation score for a restaurant in a given geospatial range. We tested our method using an available dataset. The experimental results confirmed the effectiveness of the proposed method.

Contents

Listing of figures

Listing of tables

It always seems impossible until its done.

Nelson Mandela

1

Introduction

Nowadays, the popularity and the influence of Location based social networks (LBSN) are increasing enormously. Social media such as Facebook, Foursquare, Twitter, Google Plus are providing the opportunity to share users' valuable experiences, thoughts and taste experiences with their friend's and family. An user leaves a check-in status update along with some comments or reviews about restaurants in these social networks so that others can be aware of that restaurant. By utilizing these valuable information from wide number of customers, an automatic recommendation system can built for suggesting unvisited restaurants to the interested users. Many significant work have been done related to Location based recommender systems. As suggested in [22], User's activities are time-oriented which means, often, their movements are influenced by time-period. So, they developed a collaborative recommendation model to incorporate temporal behavior and used geographical information by applying the Bayes rule for finding his next decision by using user's check-in history.

A good number of work have been done by [2], In their method they consider many aspects for serving the user. Firstly, they calculated Social Knowledge i.e. they learned the user expertise and venue popularity by applying HITS. Secondly, they project a user's history onto a previously defined category hierarchical tree. After that, by applying TF-IDF onto each node of this tree, they learn the user's preferences. Lastly, they generate their recommendation using traditional collaborative filtering to rate locations by the help of their expert users. Here, they count the tips or comments left by the user.

Another venue-centric and Time-aware suggestion work has been done by [8]. First, they calculated the popularity of a venue by measuring the volume of attendance to a venue. After that they forecast the future by applying the state of the art time series forecasting models mentioned in [11]. They also count the regular Foursquare check-in data.

Location based recommendation system is a general term. Though, there could be different of types. Such as, for movie theater, shopping male, points of interest, restaurants and so on. One of the most important points is, searching for a shopping mall is different than searching for a restaurant. Despite the fact, there are many similarities, yet they have many dissimilarities as well. Therefore, among all of these options, we choose Food recommendation system for our research. Now, there many important are factors associated behind food related venue. For example, the distance of the user from a venue is important as many prefers closer one. Then time plays a very important role as well. Generally, his breakfast menu is different than his dinner menu. Finally, popularity of a venue and user's preferences are most important for all most all of the users.

Now what we have seen so far, all the countless research works have been focused on different aspects of LBSN. Someone proposes recommendation system on user's preferences and popularity of the venues. On the other hand, others focus on the user's temporal behavior and geographical information. No one is considering all these factors at the same time. Another important point is, they are measuring all of the user's contribution whether it is check-ins, tips or comments in the past or in the present as same. Although, in terms of implementing a Food recommendation system, such measurement could cause a problem. As because, both the taste and experience of a user can change over the time. For example, a user used to like Burger in the past, might prefer Pizza at present. For that reason, the user's past and present history need to be measured differently. Furthermore, for implementing a successful Food recommendation system, we need to consider various aspects such as, the distance of the available venues from the user, user's temporal effects to the venues, user's preferences and the popularity of the venues at the same time. Hence, according to our best knowledge, no method has been proposed considering all of these key factors at the same time. Therefore, in this paper, we propose a comprehensive method for generating recommendation on Food related venues which make its final suggestion based on these four key points. Additionally, in our model, we treat all the user's historical data differently using the Logistic regression model. Logistic function provides value between 0 to 1. Finally, this paper presents the fowling contributions: 1) Firstly, we model the user's historical data in such a way so that the most recent check-in of a user gets 1 or close to 1 and the score decreases as the data is getting older. Later on, based on this data we calculate the user expertise and venue popularity. It is very common that a restaurant does not serve quality full food as it used to. Therefore, the popularity of a restaurant depends on the recent service rather than past history. In order to incorporate this idea, Logistic function helps us a lot. It counts the recent history more valuable rather than the past history. 2) Secondly, users tend to move closest venues in most cases. So, we use Logistic function which counts the distance value. By this process, closest venues are getting more score i.e. closer to 1 rather than the distant venues. 3) Thirdly, venue popularity is not only depended on the quality but also how much the venue is visited by expert users. Alternatively, the user's expertise depends on how frequently the user visits quality full venues. Moreover, a user could lose his expertise though, he might have been an expert one in past. Therefore, such user should not be counted as an expert anymore. Same goes for those venues which are not being visited by the experts anymore. In order to incorporate this mutual reinforcement relationship, calculate the user's expertise and venue popularity by using our results of logistic value mentioned first point. 4) Fourthly, User's tastes and preferences for foods are not always the same and might change frequently. For, serving the users his according to his recent preference as much as possible, we use logistic model and save his preferences category-wise. We divide user's preference into 415 different categories. By this process, user's most recent preferences count more valuable than past preferences. 5) Fifthly, Time is a very important factor in terms of food. People usually prefer different types of food breakfast, lunch and dinner. Therefore, we calculate a rating for which venue so that we could know which venue is suggest when. For serving the right venue for the right time, we divide a single a day into 6 slots where each slot represents 4 hours. We count the user's check-in frequency to each venue and save it to a particular slot from the check-in time. 6) Finally, by considering the all the five factors equally mentioned above, we generate the final recommendation for a particular user.

1.1 Motivation

Due to the popularity of internet and social media, data science has become very important and challenging sector nowadays. Location based social media has also opens the possibility to study human movement, their preference and behavior. As location based social network such as Foursquare has become very popular around the world and it provides us raw check-in data of the population from different cities, it opens the possibility for making good recommendation framework. In our thesis work we decided to make a recommendation system for suggesting restaurant using check-in data for Foursquare. From these check-in data we calculated the popularity of different restaurant using mutual reinforcement relationship between user and restaurant. Recommendation system is an emerging research sector. Many researches has been done on different recommendation systems. In our thesis work we got the chance to know about different recommendation technique. Our recommendation framework uses social knowledge to discover popularity of different venues and it is time, distance and preference aware so that it can deliver good restaurant recommendation.

1.2 Thesis Organization

The thesis organization is as follows. In Chapter 2, we present the literature review. Chapter 3 presents the detail discussion of the proposed location, time and preference aware restaurant recommendation system. The experimental results are presented in Chapter 4. Conclusion and Future works are delivered in Chapter 5.

We should not give up and we should not allow the problem to defeat us.

A. P. J. Abdul Kalam

2

Literature Review

BECAUSE OF THE INCREASING POPULARITY of location based social networks (LBSN), many studies have been done on point of interest (POI) or venue recommendation. POI recommendation can be divided into a number of categories. In the following, each categories are discussed.

2.1 Collaborative filtering (CF) Based Recommendation

Collaborative filtering (CF) is one of the widely used approach for recommendation system.CF can be divided into two categories, namely memory-based and model based. Memory-based CF can be further divided into two categories. User-centric and venue-centric. User-centric CF [2, 9, 21] focuses on details of the user profile and suggest venues based on similarity between user preference. User-centric CF first finds similar users based on their visited locations using similarity measures for example cosine similarity. In this paper [19], Badrul showed that venue-centric model gives far better performance than user-centric models and also provides better quality recommendations. One drawback of user-centric CF approach is that it requires details profile of users. On the other hand, venue-centric CF [10, 12] tries to predict probability of users may visit to a venue by investigating similarities of venues thus is does not require rich user profile.

2.2 Personalized Recommendation

In [14, 18], the authors solve personalization problem by simply asking users to manually specify their personal interests, then from these personal preferences recommendation of POIs are made. As user's preference has certain gradual and preference varies user to user, simply asking users about their preference does not give good result. They also do not consider other people's opinion. They are not much inefficient. Other new approaches try to infer user's preference from their previous visited location history. In these [7, 15, 21] papers user preference was learned by forming a user- location matrix where each row in the matrix denotes number of visits and each column denotes venue. User based Collaborative filtering (CF) is used to infer user's preference. Similarity between two user's preference is calculated by Cosine similarity function. But only CF approach does not solve the problem as data sparseness problem arises. To reduce the data sparseness problem, [16, 24] used Single Value Decomposition to a user-location matrix but this method does not work well when there is no overlap between users' location histories. Hyung in his paper [1] solved the cold-start problem when few data are available using his new approach of heuristic similarity measure.

2.3 Model Based Recommendation

Apart from techniques which deals with user's preference, other recommendation system encapsulate social opinion to determine most popular venues in a city. Model based CF builds complex model of the check in data from LBSNs and uses these models for recommendation. Ji Bao [2] introduced local expert on different categories to suggest point of interests. In his work he builds model of user expertise and venue popularity using HITS algorithm and also constructed weighted category hierarchy (WCH) of user preference based on user's location history. In this paper Y. Zheng [25] mines the most interesting locations and travel sequences from a large user-generated GPS trajectories dataset. From user-location matrix, a

HITS-based inference model was also proposed to predict the interest level of different location and also knowledge of a user. Xin Cao [3] further extends his work by considering the correlation between locations when extracting location interest level and user knowledge using HITS algorithm. But both work do not consider different location categories when doing the inference. Thus different category location and users influence each other's interest and expertise value. In our work we take the consideration of different categories during the inference.

2.4 Time-Aware Recommendation

As earlier research work [6, 17, 20] showed that people's check in behavior in LBSNs reflects significant temporal pattern and they also showed that daily pattern is most prominent temporal pattern. Few researches have been done to study temporal effect for location recommendation system on location based social networks. Huiji Gao [9] has examined temporal properties on recommendation system and has used matrix factorization considering temporal effects. He used collaborative-filtering on user's temporal behavior pattern to extract similarity between user behavior pattern. Romain [8] on the other hand invested venue centric personalized recommendation system using probabilistic approach on temporal effect. He forecasts popularity of a venue over time by time series forecasting model trained by attendance frequency of a venue. Quan Yuan [22] has further investigated temporal effect on user-based collaborative recommendation. His work deals with the temporal and spatial behavior of users. They spit the day into different time-slot based on hours and extract user's preference based on their visit. To handle data scarcity, they also calculated similarity between different time slot. In our work we also divide day into different time slot but our work differs from their work as we consider examined user - venue mutual reinforcement influence over time through the use of HITS algorithm. In [23], Zhang measured LBSN user geo-social correlations using influence matrix. Using this matrix his algorithm calculated geo-social closeness between users which is used to retrieve Top-K influential events. They also propose two approximate algorithms, namely global iteration (GI) and dynamic neighborhood expansion (DNE). Chen in his work [5] used matrix factorization (MF) methods for Personalized Point-of-interest(POI) recommendation. He used geographical influence from check-in data for recommendation. from check-in data he measured geographical influence via Multi-center Gaussian Model (MGM) and fuse it with generalized matrix factorization framework.

Our recommendation system differs from the above mentioned work in the sense that no previous work has been done on investigating model based recommendation with temporal effects. Here in this paper our proposed model takes the leverage of model based recommendation and incorporates time awareness as well as distance and preference awareness for generating recommendation.

The true sign of intelligence is not knowledge, but imagination.

Albert Einstein

3

Proposed Methodology

In our proposed restaurant recommendation method, we develop a model based approach. We take the advantage of model based recommendation by extracting social opinion and popularity of different restaurants. Usually, Recommendation System is kind of a Social Network Service. Whenever, it comes to the social media, large amount of data needs to be dealt with. As because, the more data we get, the better recommendation could be generated. In every second large amounts of data are being produced by several social media services. For example, in Facebook, more than 4.5 billion likes are generated and 1.13 billion people log on to Facebook on a daily basis. In order to deal with such huge amount of data, many costly computation needs to be performed. Normally, our machines are not fully capable in doing such type of sophisticated computation. In such case, the novel approach is, some calculation needs to be done previously. Based on those previously calculated values, online recommendation can be generated in more easy and flexible way with less amount of calculations. Therefore, our proposed method consists of two major components.

- Offline calculation
- Online recommendations

3.1 Offline Calculations

In this section, we represent the offline modeling part of our method which is comprised of.

- Learning the visiting trend of each user to each Restaurant
- Learning the user's preferences
- Learning user's expertise and restaurant popularities
- Learning time awareness of the restaurants

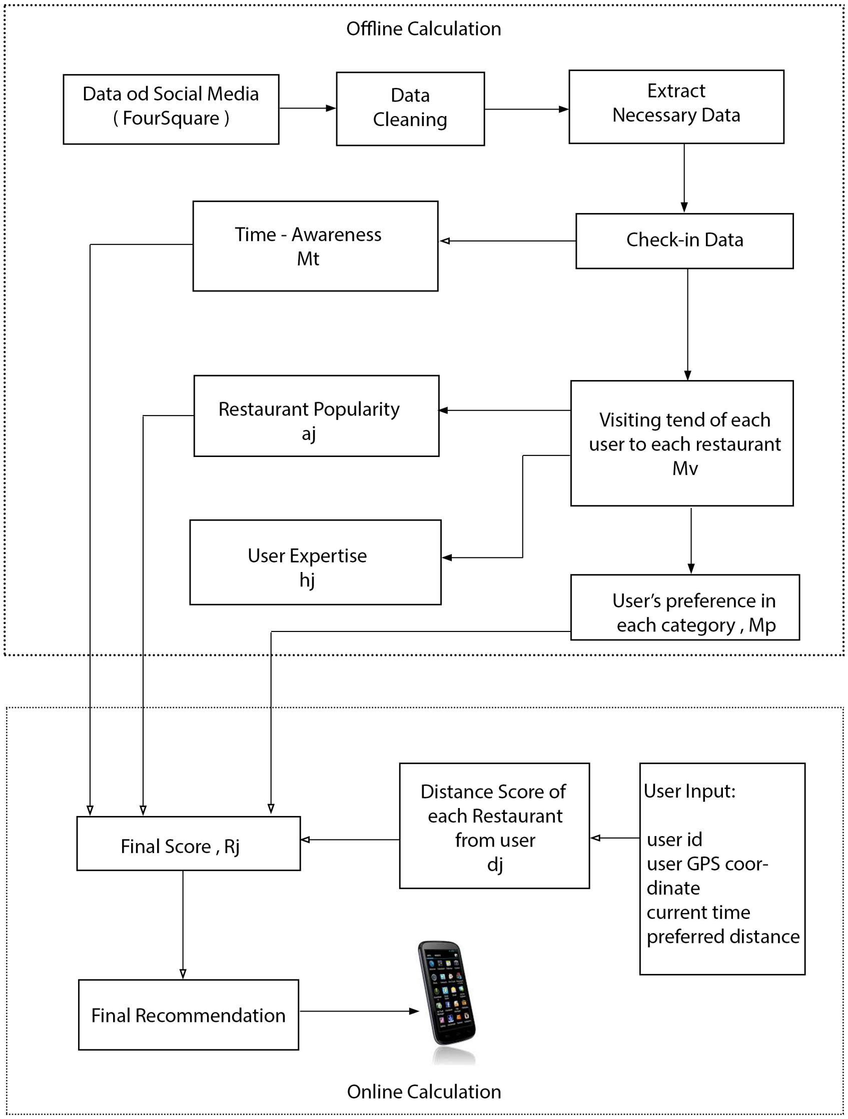

Figure 3.1: System Architecture of the proposed method.

3.1.1 Learning the visiting trend of each user to each venue

Since, last few years, Restaurant business have turned into very profitable. People are spending lots of money for food. Many popular restaurants such as McDonalds, Domino, Nandos have attracted a large number of customers. Besides of this, many new restaurants are establishing on a daily basis. A Large number of users are also visiting these restaurants. Many restaurants are indeed quality full whereas many of them are not serving well. So, it has become quit tough for the customers to choose among all the available restaurants. For that reason, a recommendation method needs to be proposed which can ensure a restaurant's quality. In order to accomplish such purpose, we use a very common intuition into our method which is measuring a user's visits to restaurants over a time-period.

Usually, a restaurant may not serve the same quality all the time. Sometimes, it might render poor service and sometimes may render satisfied service. Now, if a restaurant starts serving poor quality, then after a time period it will start losing its customers. On the other hand, if it starts serving good service to the customers, it will gain new customers instead. However, such changes do not occur immediately. Rather, it occurs gradually. Hence, the popularity of a restaurant depends on the number of user's visits it gets over a time period. The more it is visited by the users recently, the more it is considered to be as a popular restaurant in recent time and the less it is visited by users, the less popular it is. Such visits we can get via user's check-in data. In order to find a restaurant's quality, we need to consider each user's check-in differently. For such reason, the age (how old the check-in is) of a check-in is a very important factor. For example, 6-month earlier check-in, do not carry the same weight as a check-in made few days ago. Because, the restaurant may render poor service in recent time whereas before 6 months, it used to be very popular. Therefore, currently it should be rated as low. On the other hand, a restaurant might not be visited frequently by the users 6 months ago, even though in recent time, numerous visitors start visiting to this restaurant which implies that the restaurant is delivering a good service recently. Thus, the restaurant should get high rating in recent time-period. So, in order to incorporate such idea into our method, we choose Logistic function. Logistic function or logistic curve is a common "S" shape (sigmoid curve). Logistic function provides us value between 0 to 1. For example, if we map 12 months then we can get the following graph.

In Figure 3.2, we can see the function value is changing gradually. Such type of transition we use into our method whenever use the logistic function. Let, we have n users and m restaurants. A user *i* visits to restaurants *j* at time *t* is represented by $V^t_{i,j} = 0, 1$. The value is 1 if user *i* visits to restaurants *j* at time *t*. Otherwise the value is 0. Now, if a user *i* visits to restaurants *j* at time *t*, then it returns a value $w^t_{i,j}$ by using the Logistic function. Here, $w^t_{i,j}$ is between 0 and 1. The idea is, latest visit to a restaurant at time t gets more weight than the older visits.

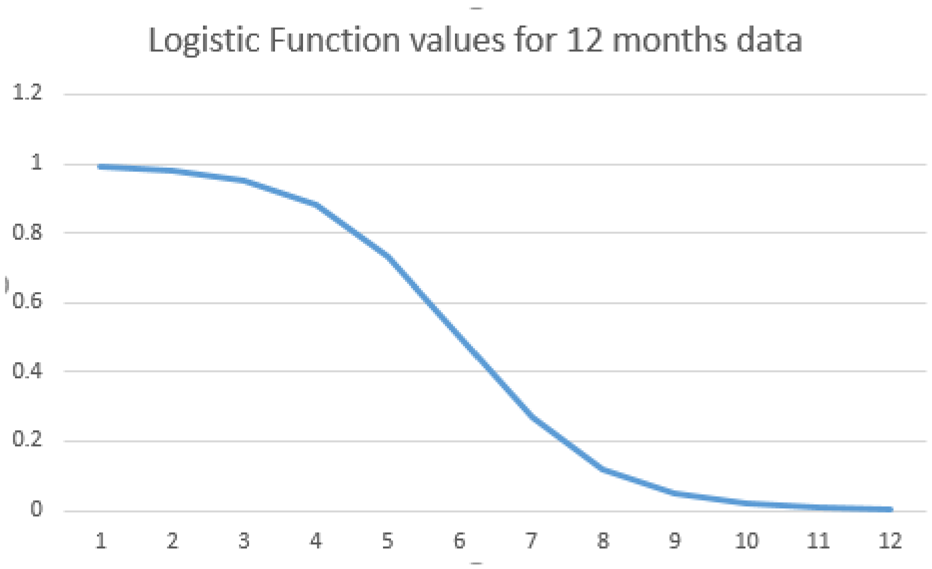

Figure 3.2: Logistic Function.

$$w_{i,j}^{t} = 1 - \frac{1}{1 + e^{\left(\frac{p}{2} - age_{v_{i,j}^{t}}\right)}}, \tag{3.1}$$

Where, p is a variable which determines the time frame in number of days and $age_{v_{i,j}^{t}}$ is an age (how old the check-in is) of a check-in inserted by user i to restaurants j at time t. Then, a matrix M_v is formed by using user's check-in history.

$$M_v = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,m} \end{bmatrix}, \tag{3.2}$$

where

$$x_{i,j} = \sum^{t} v_{i,j}^{t} \times w_{i,j}^{t}.$$

Here, each record of $\mathcal{M}_v$ is represented by $x_{i,j}$ which is the visiting trend score for user i to restaurants j.

3.1.2 Learning users' preferences

Users preference also plays a vital role in a restaurant recommendation system. Normally, user's tastes for food changes frequently. In order to serve the user best restaurant by satisfying their tastes, a user's preference learning method needs to be proposed. We can learn user's tastes by observing his or her visiting trend to each restaurant. For example, if a user used to visit Indian restaurants 3 months ago, and start visiting Chinese restaurants recently, then, we can conclude that his preference changes from the Indian food to Chinese food. Moreover, we also can infer that he loses his expertise in Indian food and becomes an expert in Chinese food because of his frequent visits to the Chinese food related venues. Thus, he is properly aware of the quality of Chinese food related venues. In terms of food, user's preferences are divided into category wise. We try to infer user's preference in each category. Categories are like Cafe, Chinese food, Indian food, Fast food and so on. Now, lets assume that we have c categories of food being served in the whole set of m restaurants, provided that each restaurant $j \in \{cp\}$. Where p represents category ID. We further summarize the values in $\mathcal{M}_v$ into another matrix

$\mathcal{M}_p$ based on their categories.

$$\mathcal{M}_p = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,c} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,c} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,c} \end{bmatrix}, \tag{3.3}$$

where

$$c_{i,p} = \sum x_{i,j} | \forall j \in \{c_p\}$$

where $\mathcal{M}_p$, represents n number of user's preferences for c number of categories individually.

- i = each user
- p = each categories
- c = total number of categories
- n = total number of users

3.1.3 Learning user's expertise and restaurant popularity

There are two types of users in a restaurant recommendation system. One type is, foodaholic people who frequently visit several restaurants and carry the actual knowledge about different restaurants. There is another type of people who are not really fond of foods and visit food related venues occasionally. Such users also have expertise but not that much compare to the first type of users. Therefore, we can conclude that, a restaurant's popularity depends on how many expert users visit that restaurant and user's expertise depends on how many popular restaurants the user visits. However, such expertise frequently gets changed. For implementing a successful recommendation method, user's most recent expertise is more valuable than his past expertise. For instance, an expert user may frequently visit a restaurant in recent time. So, we can infer that he has proper knowledge about the current condition of that particular restaurant. However, if the same user visited the restaurant 6 months ago, in that case, the quality of that restaurant may change in between that time-period. Hence, the checking of a user in past is less valuable than his checking in recent time period. Therefore, we calculate the user's expertise and restaurant's popularity by using such idea.

In our method, we first partition all users' location histories by cities as a user's knowledge usually varies in terms of geographic location. For example, a user of Dhaka City may have no idea about the restaurants in Paris. Moreover, users may have different expertise in different categories. For instance, user likes "Chinese food" in the city does not necessary have much knowledge about "Italian food" there. We learn the mutual reinforcement relationship between user's expertise and restaurant popularity by applying HITS (or Hypertext Induced Topic Search) based inference model [4, 13] to previously computed matrix $\mathcal{M}_v$.

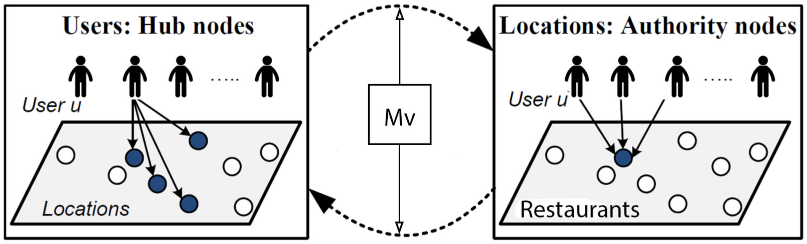

Figure 3.3: Learning Mutual relationship between user's knowledge and restaurants

$$a_j = \sum_{i=1}^{n} x_{i,j} \times h_i \tag{3.4}$$

$$h_i = \sum_{j=1}^{m} x_{i,j} \times a_j$$

Each user i, has a hub score h_i denoting its knowledge and each restaurant j is associated with an authority score a_j indicating its popularity. As a result, as shown in Equation 4, a user's knowledge can be represented by the sum of the authority scores (popularity of a restaurant) of the restaurants visited by the user and the user's visiting trend to all those restaurants. Additionally, the popularity of a restaurant can be represented by the sum of the hub scores (or knowledge) of the users visited that restaurant and visiting trend of those users to that restaurant. Hub scores are finally measured against each category. Using a powerful iteration inference method, we generate the final scores for each user and each restaurant.

3.1.4 Learning time awareness of the restaurants

Usually, our meal time is divided into three main groups. Such as breakfast, lunch and dinner. Besides, we have many more categories for example, evening food. Taking meal is also related to the local culture and tradition and varies country to country, city to city. Therefore, restaurants are often time driven. Different restaurants offer different foods based on the time of the day. Normally, restaurants serving breakfast are different than restaurants serving dinner. Therefore, suggesting correct restaurant at correct time is a very important part for a restaurant recommendation system. For such reason, we include time-awareness into our proposed method.

In order to incorporate the time awareness in the proposed method, we analyze the frequency distribution of the check-in history of a restaurant. 24 hours of the day is equally divided into 6 bins and the visits are normalized, assuming that $sf_{j,b}$ represents total number of check-ins to a restaurant j for bin b. tf_j represents total number of check-ins for restaurant

Bin ID	Time Frame
1	12 am - 4 am
2	4 am – 8 am
3	8 am – 12 pm
4	12 pm – 4 pm
5	12 pm – 4 pm
6	8 pm – 12 am

Table 3.1: List of Time bins

j.

$$M_t = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,m} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ f_{6,1} & f_{6,2} & \cdots & f_{6,m} \end{bmatrix}, \tag{3.5}$$

where

$$f_{b,j} = \frac{sf_j^b}{tf_j}.$$

Now, visiting frequency distribution for a Coffee-Shop, a French Restaurant, a BAR and an Italian Restaurant are given in the following figures. Where,

Usually, all the offline calculations need to be done once in a day or once in a week depending on the system owner's requirements. Therefore, all the matrices such as M_v, M_p and M_t will be updated on a regular basis. By doing so, we can analyze user's most current behavior and compare them with their past experiences so that we can always generate recommendation based on the most current and updated trend.

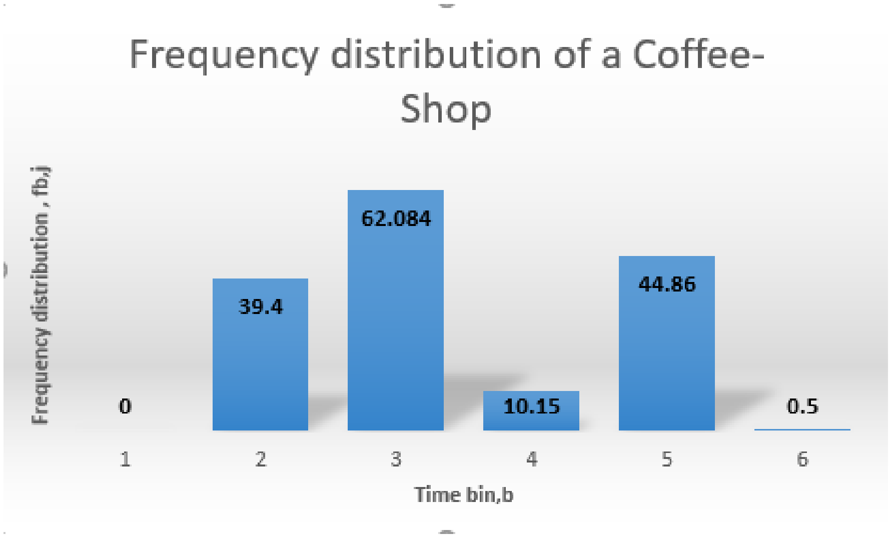

Figure 3.4: Frequency Distribution of Customers Visit to a Coffee Shop

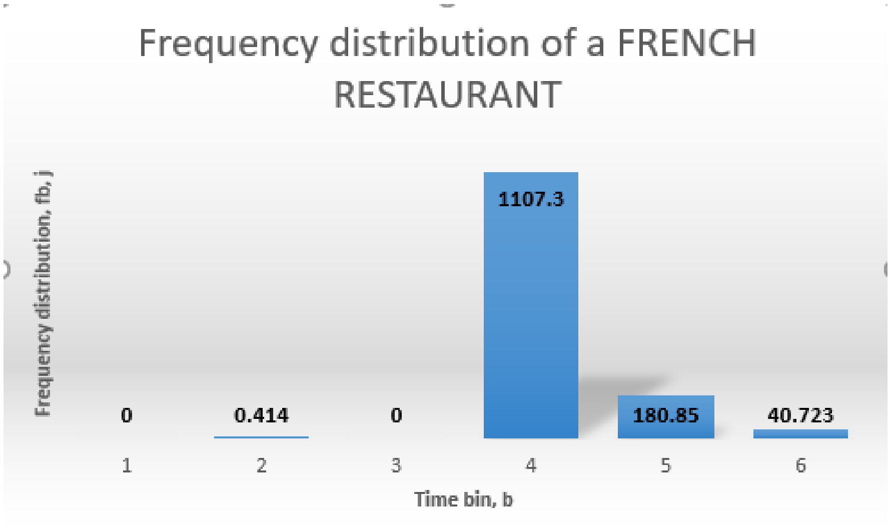

Figure 3.5: Frequency Distribution of Customers Visit to a French Restaurant

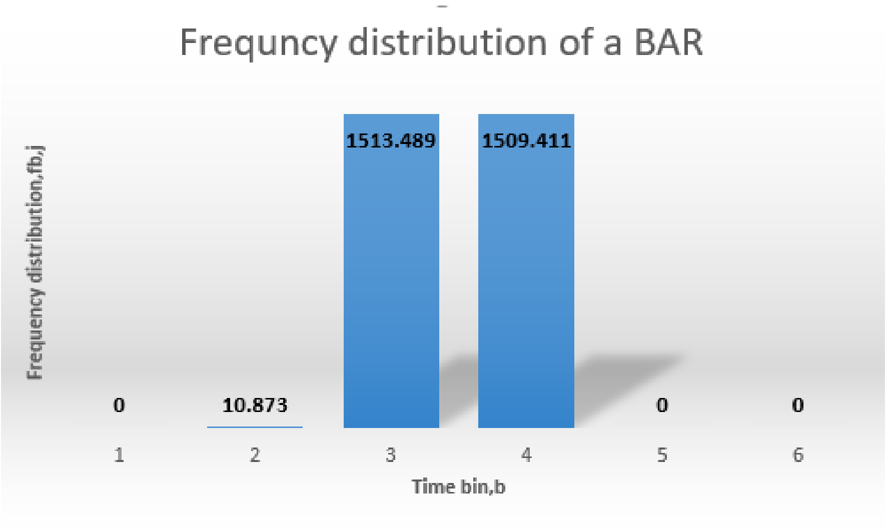

Figure 3.6: Frequency Distribution of Customers Visit to a BAR

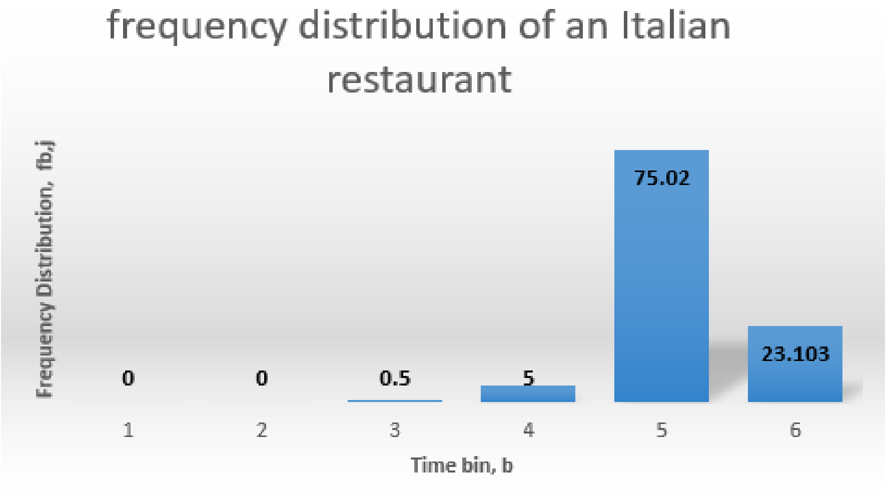

Figure 3.7: Frequency Distribution of Customers Visit to a Italian Restaurant

3.2 Online Recommendation

The Online calculation are done whenever the user request for recommendation. Light weighted calculation needs to be done for rendering the request.

In the proposed method, a user device needs to provide the following information for online recommendation.

- User's ID
- User's GPS coordinate
- Current time
- Preferred distance, pd
- Number of suggestions, K

Here, first three inputs taken from the user's device automatically. The 4th and the 5th one needs the manual insertion. After all of these inputs, final recommendation is generated. Our Online computation contains two major components.

- Scoring the restaurants based on the distance form the user
- Generating final recommendation

3.2.1 Scoring the restaurants based on the distance form the user

Users always tend to visit closer restaurants. Though some of the users prefer distant restaurants. Nevertheless, such event occurs barely. Specially, when they are out for a tour or for some recreation purpose. In real life, large amount of recommendation requests come from the users at their working place or users who are in urgency. In such case, they normally prefer closer restaurants than the distant ones. Therefore, we incorporate the idea of serving closer restaurants to our users into the proposed method. In order to do that, all the restaurants j within the user's preferred range are identified and are assigned weight values dj using the logistic function. The value is between 0 to 1. As a result, the restaurants closer to the user, gets value of dj closer to 1 and decrease gradually as the distance $dist_j$ increases.

$$d_j = 1 - \frac{1}{1 + e^{\left(\frac{pd}{2} - dist_j\right)}}, \tag{3.6}$$

where pd is the preferred distance, $dist_j$ is the distance of a restaurant from the user's current location.

3.2.2 Generating final recommendation

Now, we prepare all the necessary components for generating the final recommendation. Our Final recommendation is based on four major components.

- Distance score of each restaurant, d_j

- Time-awareness score of each restaurant, $f_{b,j}$
- The popularity score of each restaurant, a_j
- User's preference score on each restaurant based on the restaurant's category, $C_{i,p}$

At last, the recommendation score for a restaurant R_j is generated by using the following equation.

$$R_j = d_j \times f_{b,j} \times a_j \times c_{i,p} | j \in \{c_p\} \tag{3.7}$$

Finally, the users are being recommended top k restaurants based on R_j values.

Believe you can and you're halfway there.

Theodore Roosevelt

4

Experimental Results

In this chapter, we discuss about the experimental results.

4.1 Dataset

Our dataset includes long-term (about 18 months from April 2012 to September 2013) global-scale check-in data collected from Foursquare. It contains 33,278,683 check-ins by 266,909 users on 3,680,126 venues (in 415 cities in 77 countries). Those 415 cities are the most checked 415 cities in the world, each of which contains at least 10000 check-ins. The Check-ins table contains all check-ins with 4 columns, which are.

1. User ID (anonymized)
2. Venue ID (Foursquare)
3. UTC time
4. Time zone offset in minutes (The offset in minutes between when this check-sin occurred and the same time in UTC, i.e., UTC time + offset is the local time)

Point of Interest table contains all venue data with 5 columns, which are.

- Venue ID (Foursquare)
- Latitude
- Longitude
- Venue category name (Foursquare)
- Country code (ISO 3166-1 alpha-2 two-letter country codes)

City table contains all 415 cities data with 6 columns, which are.

- City name
- Latitude (of City center)
- Longitude (of City center)
- Country code (ISO 3166-1 alpha-2 two-letter country codes)
- Country name

- City type (e.g., national capital, provincial capital)

We calculate and construct all of our Matrices and tables based on these datasets. Basically we form many Matrices but the mentioned above are main. For example, Point of Interest table has venue's latitude and longitude and City table has city center's latitude and longitude on the other hand. However, we need to know which venue belongs to which city and that information is absent. Therefore, we calculate which venue belongs to which city from these two tables. Next, we choose a city for which we implement and test our recommendation engine and we run some test case for that. Like which city has more diverse types of food related venues. Finally, we chose Paris for our experimental purpose. First, we extract all the check-ins for Paris which is total 111325. Secondly, we find total 7991 number of restaurants and 6903 users for Paris. Then, using these datasets we do rest of our calculation.

4.1.1 System description

Our system is implemented in a Personal Computer having processor Intel(R) Core I3 CPU @ 2.10 GHz, 6.00 GB RAM and 64-bit windows Operating System. The whole system is implemented Python 2.7. For our project, we use very popular python packages.

- Numpy
- Scipy
- Pandas
- Jupyter Notebook

4.2 Generating Final Results

4.2.1 Matrices and Vectors

We have fully implemented our proposed method. At first, we form M_v matrix from the data Check-ins table. After that, by summarizing the M_v, we form two new matrices M_p and M_t. We also form two new matrices M_a and M_h and store the values of a_j and h_j in them. These are our offline computation. After getting the user's preferred range pd, we for a vector $v_{d,j}$, where each value is d_j.

4.2.2 Final Results

The following is a sample input.

- user_id = 1455
- user_GPS_coordinate = 0.852728, 0.039689
- current_time = 7:35 am
- distance_range = 5 Km
- Number of suggestion = 11

In this suggestion, the first suggested restaurant is situated within 0 kilometers which is very close. Secondly, we can see the restaurant is almost perfect for the time when the request is being made. Thirdly, the category of the suggested restaurant is not user's top preference but carries his average preference. Fourthly, the suggested restaurant is the most popular

j	C_p	Distance	d_j	$f_{b,j}$	$c_{i,p}$	a_j	R_j
V1	Coffee Shop	0.002036	0.923998988	0.914894	0.238406	7887.975	1589.734
V2	Fast Food Restaurant	3.694498	0.232455524	0.111111	3.460089	501.5836	44.82579
V3	Coffee Shop	3.789084	0.216007915	0.352113	0.238406	603.0691	10.93542
V4	Coffee Shop	4.815266	0.089866505	0.314286	0.238406	411.366	2.769924
V5	Fast Food Restaurant	1.22496	0.781604266	0.038462	3.460089	25.94135	2.69832
V6	French Restaurant	1.76605	0.675671489	0.090909	0.238406	176.137	2.57935
V7	Coffee Shop	2.974756	0.38349122	0.125	0.238406	209.8358	2.398071
V8	Bakery	2.410427	0.522378367	1	2.556918	1.366651	1.825407
V9	Bakery	1.202104	0.785480611	0.2	2.556918	3.085932	1.239563
V10	Fast Food Restaurant	1.454289	0.73995047	0.02381	3.460089	17.68638	1.078151
V11	Coffee Shop	3.692519	0.232808775	0.153846	0.238406	109.7478	0.937127

Table 4.1: Recommendations for user 1455

restaurant among all of other available options. Therefore, our method predicts that the user might want to pay a visit to this restaurant. Then, our second suggestion is not that much close but within the user defined range. Secondly, the restaurant is less visited during the input time but one important point is to mention that our time score is restaurant centric not the category centric. This restaurant has such score because the restaurant might be new. Although, Fast Food restaurant is totally relevant during the input time. It would be odd if a restaurant which serves dinner suggested instead. Next, the restaurant matches with the user's preference. Lastly, the restaurant is very popular. So, based on these factors the restaurant is suggested at second position.

Now, one might think that our method has become authority or popularity score centric due to very amount of score saved in them. However, that is not the case. For example, let us analyze the two restaurant suggested at 10th and 11th position. The suggested restaurant

j	C_p	Distance	d_j	$f_{b,j}$	$c_{i,p}$	a_j	R_j
V1	French Restaurant	0.000708	0.924092	0.369697	7.991553	102.4333	279.6621
V2	French Restaurant	1.400092	0.750243	0.404255	7.991553	92.15041	223.3501
V3	French Restaurant	0.40745	0.890177	0.243902	7.991553	97.5733	169.2991
V4	French Restaurant	1.287799	0.770688	0.419355	7.991553	61.10506	157.8224
V5	French Restaurant	2.281554	0.554395	0.466667	7.991553	63.26967	130.8137
V6	Cafe	3.198533	0.332138	0.294118	2.721516	479.6809	127.5271
V7	Bar	6.144681	0.025464	0.446429	2.177268	4040.685	100.0119

Table 4.2: Recommendations for user 2273

at the 10th position has less popularity score but very high user's preference. On the other hand, the restaurant at 11th position is very popular compare to the previous one. But, does not match with the user's preferences.

Next let us analyze for another user, 2273.

- user_id = 2273
- user_GPS_coordinate = 0.852734, 0.039656
- current_time = 5:35 pm
- distance_range = 5 km
- Number of suggestion = 7

Now for this user input, let us analyze our final recommendation. For the first suggested restaurant, we can see the venue is very close to the user's position. Secondly, the venue has time score not that much but not awfully low. Thirdly, the venue category is highly preferred by the user. Fourthly, the venue is very popular. First five suggestion has similar sort of scores for different factors. Now, the venue V7 is poorly suggested despite of high popularity as the restaurant is out of the range of user specified range.

5

Conclusion

In this paper, we demonstrate a Location based Time-aware and Preference-aware Restaurant Recommendation Method. At first, using the Logistic function, we calculate user's visiting trend to each restaurant. Then, using this information and utilizing the concept of HITS, we calculate the authority score of each restaurant and hub score for each user in each category. Next, we calculate the user's preference in each category using the same visiting trend information. Lastly, we divided each restaurant into 6 groups where each group keeps the record of visiting frequency for four hours of a particular restaurant so that we could infer which restaurant to suggest when. Such approach makes our method time-aware. All these are our offline calculation. For online recommendation, we take user's coordinate position, time, user's preferred range in kilometer and user id as input. Based on such information, we generate our final recommendation which is based on four key factors. 1) From the coordinate position and user defined range, we assign each restaurant around the user a distance score, using logistic model. 2) From the current-time input, we calculate a time score for each restaurant using our time-aware offline calculation. 3) We measure the user's preference score for each restaurant which is category dependent. 4) We allocate each restaurant's popularity score. Finally, by combining all these four key features and influences, we generate our final recommendation.

5.1 Future work

As we have proposed and tested a Location based Time-aware and Preference-aware Restaurant Recommendation method, we think there are some rooms for improving the method. Firstly, we want to take more control over the four key factors so that we could decide which factor should contribute how much influence over the final recommendation for a particular user. Although, such type control depends on the system owner requirements. Secondly, at this modern era, Social media is a part and parcel of our daily life. Even though, we use the data-set of Foursquare, other services like Facebook, twitter and Instagram are very popular as well. In future, one of our main goals is to incorporate the influence of these social networks into our system. Thirdly, the method is a proposed method. Therefore, it needs to be tested with several other methods in order to measure the performance in a more precise way.

References

[1] Ahn, H. J. (2008). A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences*, 178(1), 37–51.

[2] Bao, J., Zheng, Y., & Mokbel, M. F. (2012). Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems* (pp. 199–208).: ACM.

[3] Cao, X., Cong, G., & Jensen, C. S. (2010). Mining significant semantic locations from gps data. *Proceedings of the VLDB Endowment*, 3(1-2), 1009–1020.

[4] Chakrabarti, S., Dom, B., Raghavan, P., Rajagopalan, S., Gibson, D., & Kleinberg, J. (1998). Automatic resource compilation by analyzing hyperlink structure and associated text. *Computer Networks and ISDN Systems*, 30(1), 65–74.

[5] Cheng, C., Yang, H., King, I., & Lyu, M. R. (2012). Fused matrix factorization with geographical and social influence in location-based social networks. In *Aaai*, volume 12 (pp. 17–23).

[6] Cheng, Z., Caverlee, J., Lee, K., & Sui, D. Z. (2011). Exploring millions of footprints in location sharing services. *ICWSM*, 2011, 81–88.

[7] Chow, C.-Y., Bao, J., & Mokbel, M. F. (2010). Towards location-based social networking services. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks* (pp. 31–38).: ACM.

[8] Deveaud, R., Albakour, M., Macdonald, C., Ounis, I., et al. (2015). Experiments with a venue-centric model for personalisedand time-aware venue suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (pp. 53–62).: ACM.

[9] Gao, H., Tang, J., Hu, X., & Liu, H. (2013). Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM conference on Recommender systems* (pp. 93–100).: ACM.

[10] Hu, B. & Ester, M. (2013). Spatial topic modeling in online social media for location recommendation. In *Proceedings of the 7th ACM conference on Recommender systems* (pp. 25–32).: ACM.

[11] Hyndman, R. J., Koehler, A. B., Snyder, R. D., & Grose, S. (2002). A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18(3), 439–454.

[12] John, G. H. & Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence* (pp. 338–345).: Morgan Kaufmann Publishers Inc.

[13] Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5), 604–632.

[14] Kodama, K., Iijima, Y., Guo, X., & Ishikawa, Y. (2009). Skyline queries based on user locations and preferences for making location-based recommendations. In *Proceedings of the 2009 International Workshop on Location Based Social Networks* (pp. 9–16).: ACM.

[15] Levandoski, J. J., Sarwat, M., Eldawy, A., & Mokbel, M. F. (2012). Lars: A location-aware recommender system. In *2012 IEEE 28th International Conference on Data Engineering* (pp. 450–461).: IEEE.

[16] Ma, H., Yang, H., Lyu, M. R., & King, I. (2008). Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management* (pp. 931–940).: ACM.

[17] Malmi, E., Do, T. M. T., & Gatica-Perez, D. (2012). Checking in or checked in: comparing large-scale manual and automatic location disclosure patterns. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia* (pp.26).: ACM.

[18] Park, M.-H., Hong, J.-H., & Cho, S.-B. (2007). Location-based recommendation system using bayesian user's preference model in mobile devices. In *International Conference on Ubiquitous Intelligence and Computing* (pp. 1130–1139).: Springer.

[19] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web* (pp. 285–295).: ACM.

[20] Ye, M., Janowicz, K., Mülligann, C., & Lee, W.-C. (2011). What you are is when you are: the temporal dimension of feature types in location-based social networks. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (pp. 102–111).: ACM.

[21] Ye, M., Yin, P., & Lee, W.-C. (2010). Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems* (pp. 458–461).: ACM.

[22] Yuan, Q., Cong, G., Ma, Z., Sun, A., & Thalmann, N. M. (2013). Time-aware point-of-interest recommendation. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval* (pp. 363–372).: ACM.

[23] Zhang, C., Shou, L., Chen, K., Chen, G., & Bei, Y. (2012). Evaluating geo-social in-

fluence in location-based social networks. In *Proceedings of the 21st ACM international conference on Information and knowledge management* (pp. 1442–1451).: ACM.

[24] Zheng, V. W., Zheng, Y., Xie, X., & Yang, Q. (2010). Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th international conference on World wide web* (pp. 1029–1038).: ACM.

[25] Zheng, Y., Zhang, L., Xie, X., & Ma, W.-Y. (2009). Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web* (pp. 791–800).: ACM.