

# Point-of-Interest Recommendation Systems in Location-Based Social Networks

Myeongjae Song  
Division of Science and Mathematics  
University of Minnesota, Morris  
Morris, Minnesota, USA 56267  
songx823@morris.umn.edu

## ABSTRACT

Accurate point-of-interest (POI) recommendations are key to the success of location-based social networks (LBSN) since it will attract more users and advertisers to the platform. To achieve the POI recommendation task, many methods are proposed and used in practice from generic recommendation system methods to POI specific algorithms. In this paper, we will analyze a novel recommendation technique, factorized personalized Markov chain (FPMC) model, which is a combination of matrix factorization and Markov chain models. We will also explore enhancements of FPMC optimized for POI specific characteristics such as users' movement constraint and complex behavior over time. Experimental results show POI specific methods outperform generic recommendation models. It also verifies that the more POI characteristics a system incorporates, the more accurate the prediction.

## Keywords

Recommendation Systems, Point-of-Interest Recommendation, Location-Based Social Networks

## 1. INTRODUCTION

Nowadays, location-based social networks (LBSN), such as *Facebook places*, *Tinder*, and *Yelp*, have been gaining a lot of attention with the widespread use of smartphones embedded with the global positioning system (GPS). Even though LBSNs are relatively new compared to the traditional social networking services, millions of active users use the platform on a daily basis voluntarily sharing their location information. Many of these LBSNs allow users to “check-in” to places like a restaurant or cinema; we call these checked-in places the points of interest (POI) of the users. Users can share videos, pictures, or reviews about these places via this “check-in” feature. Based on the collected user data, LBSNs provide POI recommendations. POI recommendation is very important for LBSNs for two reasons. First, users are more likely to use the platform and check-in places for their own benefit if the quality of recommendation is accurate. Second, it allows targeted advertisers to serve specific user groups. Therefore, accurate POI recommendation is crucial for the success of LBSNs. [1]

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/>.  
*UMM CSci Senior Seminar Conference, December 2017 Morris, MN.*

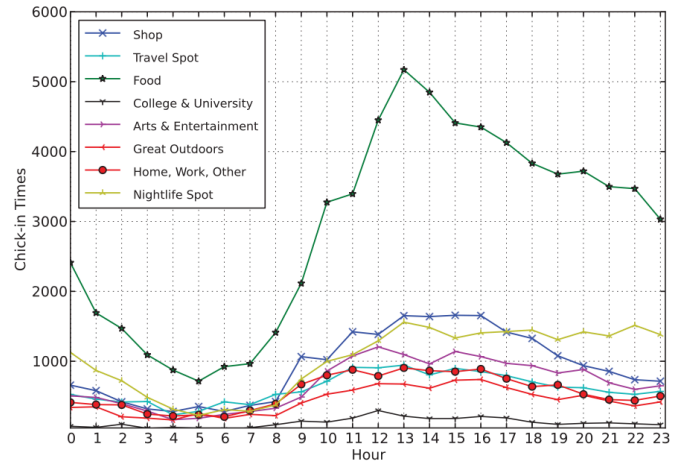
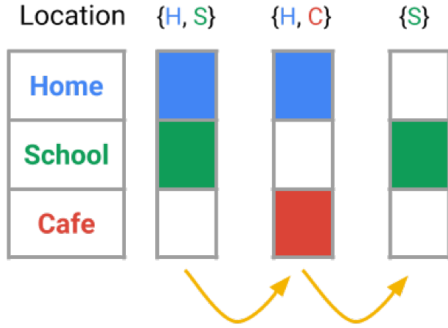


Figure 1: Relationship between time interval and check-in location category in NYC. [3]

Even though POI recommendation is a type of recommendation system, it has distinct characteristics mainly due to its geographical nature. For instance, recommending a movie is a fairly different task from recommending a place. A movie recommendation system can recommend any movie, but a POI recommendation system should not recommend a sushi restaurant in Tokyo for a user in Minnesota. The following are meaningful differences between conventional recommendations and POI recommendation:

- The types of checked-in places are highly related to the time period. Figure 1 shows check-in data in NYC from an LBSN (*Foursquare*) over different hours of a day. The number of check-ins clearly varies depending on the time.
- People are likely to visit nearby places because of geographical limitation. Not many people are willing to fly to Japan from the US just for a nice sushi restaurant. [3]
- The transition between POIs is strongly affected by the user's own preference. Li et al. [3] refer to this as a long-term individual preference. For instance, some people go to the gym after work, but some people go home right away after work.
- A user's next location is highly related to the user's



(a) Transition diagram of a Markov chain for sets.

	Home	School	Cafe	#
Home	1/2	1/2	1/2	2
School	1/1	0/1	1/1	1
Cafe	0/1	1/1	0/1	1

(b) Transition matrix for (a).

Figure 2: Markov chains for sets.

current location. We would call this the sequential feature of POI recommendations.

Traditionally, matrix factorization (MF) or Markov chains (MC) have been popular techniques for producing recommendations. The ideas behind MF and MC will be explored in more detail in section 2. Because of the known drawbacks of each model, Rendle et al. [6] propose factorized personalized Markov chains (FPMC), which is a combination of MC and MF techniques. FPMC is a robust system, but it is a generic model and not designed for POI recommendations. To capture the locality feature of POI recommendations, Cheng et al. [1] extend FPMC to include localized region (LR) information. This extension is abbreviated as FPMC-LR. And, Li et al. [3] propose time-aware FPMC with time decaying consideration called TAD-FPMC.

In section 2, we are going to cover mathematical backgrounds in recommendation systems to understand the proposed models. Then in section 3, we will explore the proposed approaches in detail. In section 4, we will compare the accuracy of different recommendation techniques: FPMC, FPMC-LR, and TAD-FPMC. Finally, we will summarize our findings in section 5.

## 2. BACKGROUNDS

**Markov chain.** A Markov chain (MC) is a stochastic process widely adapted in many recommendation systems including movie, item, and POI recommendations. An MC is defined on a set of states; each state can be either a value or a set of values. In the context of LBSN, each state represents a place where users can visit. Now, we can define an MC as a collection of transitional probabilities among different states. In other words, it is a map of transitions from one state to another state where transitions are labeled with their probability. It is worth pointing out that the sum of all transitions leading out of a state must sum to 1. Another important characteristic of MCs is that predictions of future transitions are solely based on the present system. Therefore, the previous iterations and future iterations are completely independent.

When used in location predictions, we often want each state to capture all locations visited over a specific interval of time and for our transitions to occur at fixed intervals. This will simplify calculations by allowing us to divide a day into evenly broken hours. By expanding our set

of states to be non-empty subsets of the set of locations,  $L$ , we accomplish this goal. This new expansion is called Markov chains for sets. Unlike traditional MCs, the size of the state space is  $2^{|L|}$ . Figure 2a shows an MC for sets. There are three different locations of interest *home*, *cafe*, and *school*. These would be the states in a traditional MC model. In an MC for sets, we have 7 non-empty subsets such as  $\{home, school\}$  and  $\{school\}$ . The extended transition matrix for our expanded set of states rapidly becomes too large to be manageable. MCs for sets deal with this by estimating a per-location transition matrix that can be used to calculate the probabilities of the extended transition matrix. This per-location matrix is estimated using observed transitions. Rendle et al. [6] refer to it as a transition matrix, but it is something a little bit different.

It is easier to understand with an example. There are two observed transitions in figure 2b:  $\{home, school\} \rightarrow \{home, cafe\}$  and  $\{home, cafe\} \rightarrow \{school\}$ . Let us call the location associated with a row *start-location* and the location associated with a column *end-location*. An entry in our per-location transition matrix is calculated by dividing the number of transitions containing the given *start-location* and *end-location*, by the total number of transitions containing *start-location*. This total number is indicated in 2b by the # column. For example, we put a 2 in the column for the row *home* since 2 transitions start with a set containing *home*. This number is then the denominator in all the fractions of the per-location transition matrix. There is only 1 transition from a set containing *home* to a set containing *school*, so the entry for *home*  $\rightarrow$  *school* is 1/2. This represents the probability that a state containing *home* will transition to a state containing *school*.

As illustrated in the examples above, we can imagine Markov chains are directly applied to POI recommendations but with tens of thousands of places instead of a few places. Note, in practice, MC based approaches use one general transition matrix over all users.

**Matrix factorization.** In Linear Algebra, matrix factorization (MF) is a technique to factor a matrix as a product of multiple matrices. When used in POI recommendations, the original matrix has two dimensions: locations and users. Each row represents a location and each column represents a user. Each entry in the matrix occurs at a distinct row and column; this entry represents a user’s rating for that particular location. As you can see in figure 3, the final matrix

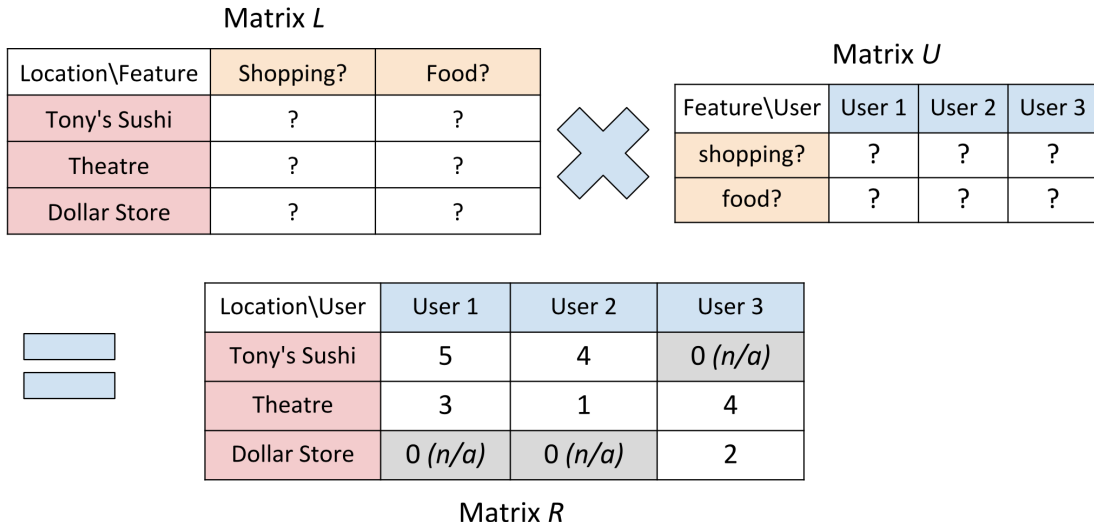


Figure 3: Matrix factorization of location-user matrix.

in the figure has many 0 entries which are the places users have never visited or not rated, so no rating exist.

Our goal is filling the empty entries by calculating predicted ratings. This is where MF becomes relevant. By factorizing the location-user matrix, it can be represented as a product of two matrices: a location matrix  $L$  and a user matrix  $U$ . Ratings matrix  $R$  from figure 3 is the product of  $L$  and  $U$ :  $R = LU$ . You might notice that  $L$  and  $U$  do not have the same dimensionality as  $R$ . For matrix multiplication to be defined, the number of columns in  $L$  and the number of rows in  $U$  must match. We call this number  $f$ ; it represents the features. In the example, I have chosen *shopping?* and *food?* for simplicity, but in general the algorithm that does the estimating is given a value for how many features to include and there is not always such a clear-cut way of explaining their meaning. In the diagram, the given number of features is 2. Once we find the two factors of  $R$ , we can make an estimation of unknown ratings. Let  $q_l$  be the row associated with location  $l$ , and  $p_u \in \mathbb{R}^f$  be the column associated with user  $u$ . The elements of  $q_l$  show how related location  $l$  is to its features, and the elements of  $p_u$  show how interested user  $u$  is in the features. If  $r_{u,l}$  is the rating of location  $l$  from user  $u$ , we can find the estimate rating via multiplying corresponding entries. [2]

$$r_{u,l} = q_l p_u \quad (1)$$

The following are some clarifications on the estimation process:

- The location-user matrix has many missing entries. Thus, the matrix factors ( $L$  and  $U$  from figure 3) are estimates whose product produces a matrix containing entries similar to the original matrix.
- The size of  $f$  matters with respect to accuracy. If  $f$  is too small, we do not incorporate enough features for precise predictions. Koren et al. [2] verified increasing the size of  $f$  results in more precise predictions.

The factors can be estimated using machine learning techniques. Rendle et al. [5] use a technique known as Bayesian

personalized ranking (BPR) that attempts to produce factors that will be useful for providing a personalized ranking for the locations. To apply BPR to Markov chains for sets, sequential BPR (S-BPR) is used. The details of BPR and S-BPR are beyond the scope of this paper. For more details, see [4] and [5].

**Tensor.** From our point of view, a tensor is a multidimensional array in any programming language. The rank of the tensor is then the number of components needed to specify an entry in the array. A vector is a tensor of rank 1 and a scalar is a tensor of rank 0. A rank 2 tensor can be represented as a matrix, and higher rank tensors are thought of as a multidimensional version of a matrix. For instance, we can stack matrices on the top of one another to represent a 3 rank cube-shaped tensor. Just like matrices can be factorized, tensors can also be factorized. This is a key idea of many POI recommendation approaches that we will discuss more in the next section.

### 3. FACTORING PERSONALIZED MARKOV CHAINS AND ITS IMPROVEMENTS

As we have discussed in section 2, MC and MF based recommendation systems have fairly distinct characteristics from each other. In this section, we will study a model called factorized personalized Markov chains (FPMC) that incorporates MF into MC, and its enhancements specializing in LSBN specific data properties.

#### 3.1 Formalization

Before we dive into the details of the algorithms, let us introduce the notation used in this paper. Let  $U$  be a set of users and  $L$  be a set of locations. For each user  $u \in U$ , we know their check-in history set  $L_u^t$  where  $t$  is the interval of a user visit. Our goal now is, given the user check-in data  $L_u^1, \dots, L_u^{t-1}$ , recommend a next location to a user  $u$  at time  $t$ . This goal is roughly equivalent to finding the probability of  $u$  checking in location  $l$  at time  $t$ , given a previous check-in location  $i$  at time  $t - 1$ :

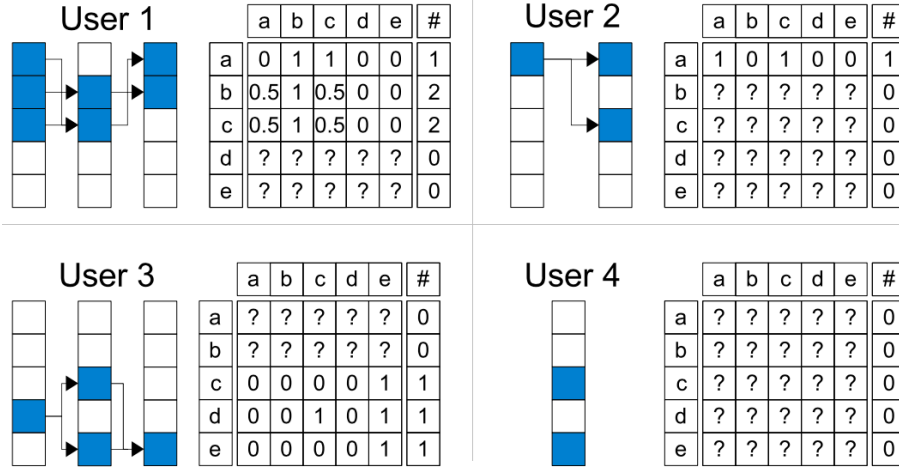


Figure 4: Personalized Markov chains. [6]

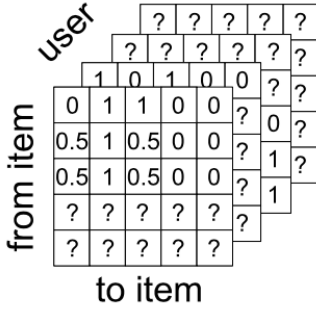


Figure 5: Personalized transition tensor. [6]

$$x_{u,i,l} = p(l \in L_u^t | i \in L_u^{t-1}) \quad (2)$$

### 3.2 FPMC

Even though MC based recommendation systems were widely studied and used because of its capacity to capture sequential data, it lacks the personalization feature to distinguish each user. MF also has its weakness in processing sequential information since the order in a user’s check-in sequence is often ignored when the prediction is calculated. In section 1, we mentioned sequential information plays a huge role in LBSNs in that there is a strong connection between users’ recently visited locations and future locations. Imagine a POI recommendation system with no sequential awareness. It may recommend a family restaurant to a user who just checked into a restaurant; this is not ideal. To take advantage of these two complementary models, MC and MF, Rendle et al. [6] propose FPMC.

First, let us think about what kind of recommendation model we aim to build. We want a recommendation system that is aware of sequential pattern and personalized for each user. MC is good at recognizing sequential features, but it uses one general transition matrix for every user. Then, the question is, can we create a personalized MC? A straightforward approach is having a personalized MC per user  $u \in U$ .

Figure 4 shows four different transition matrices for each user. The question mark (?) entries mean we have no data to estimate the probability of the transition. The problem of these personalized MC matrices is that they are often very sparse, so they result in poor estimations. We just do not have enough data from each user. This is the place where the ideas of MF can be applied. By stacking all transition matrices of users, we can get a transition cube or tensor  $\chi$ . Now, we can factorize the tensor using a tensor factorization technique. Factorizing this tensor  $\chi$  produces four parameters: one rank-3 core tensor and three matrices. Using an efficient tensor factorization technique pairwise interaction tensor factorization (PITF), we get the following estimation equation for the probability of user  $u$  going to location  $l$  from location  $i$ . [7]

$$\hat{x}_{u,i,l} := \underbrace{v_u^{U,L} \cdot v_l^{L,U}}_A + \underbrace{v_l^{L,I} \cdot v_i^{I,L}}_B + \underbrace{v_u^{U,I} \cdot v_i^{I,U}}_C \quad (3)$$

- A: Interaction between the user features  $v_u^{U,L}$  and the next location  $v_l^{L,U}$ .
- B: Interaction between the next location  $v_l^{L,I}$  and the current location  $v_i^{I,L}$ .
- C: Interaction between the user features  $v_u^{U,I}$  and the current location  $v_i^{I,U}$ .

This is the essence of the FPMC method. To achieve the recommendation goal  $\hat{x}_{u,i,l}$ , we construct a tensor containing MCs for users, and then we factorize the tensor into a rank-3 tensor and matrices. Finally, we calculate the prediction by combining the corresponding entries of the factors.

### 3.3 FPMC-LR

FPMC successfully constructs a personalized MC model for POI recommendation by combining ideas of MF and MC. However, it overlooks another noticeable property of an LBSN dataset, localized region constraint. According to check-in data from *Gowalla* and *Foursquare*, more than 75% of check-ins from *Foursquare* and 80% of check-ins from

*Gowalla* happened within 10 km of the previously checked-in place. [1] The observation on the LBSN data shows the trend that users tend to check in places close to their previous check-ins, but FPMC does not really make use of this potentially useful trend.

To add the ability to use localized region constraint information into the existing FPMC model, Cheng et al. [1] introduce FPMC-LR which combines the FPMC model with localized region (LR) constraints to provide accurate successive POI recommendation in LBSNs. Because FPMC-LR only takes into account nearby candidate locations depending on where the users currently are, it reduces the computational costs and noisy information. Also, it results in more accurate predictions as we will see in section 4.

FPMC-LR has a similar setting to FPMC since it is an improvement over FPMC. Just like FPMC, the fundamental goal of FPMC-LR is to give the most suitable location recommendation for user  $u$  at time  $t$  given a sequence of check-in data. In other words, we need to calculate  $x_{u,i,l}$ , which is the probability of a user  $u$  visiting a location  $l$  at time  $t$ , where  $i$  is the user location at time  $t-1$ . The main difference between FPMC and FPMC-LR is in their transition tensor. As we saw earlier, FPMC takes into account all possible locations for each user, so the tensor looks like this:

$$\chi \in [0, 1]^{|U| \times |L| \times |L|} \quad (4)$$

Unlike FPMC, FPMC-LR only considers neighborhood locations of the previous check-in location  $i$ . The tensor transition in FPMC-LR looks like this:

$$\chi \in [0, 1]^{|U| \times |L| \times |N_d(L)|} \quad (5)$$

The set of neighborhood locations  $N_d(L)$  is calculated by the Haversine formula, which is used to find the shortest distance between two points on the surface of a sphere. [1] In this case, we assume the earth is roughly a sphere. By applying the same tensor factorizing technique PITF to our tensor  $\chi \in [0, 1]^{|U| \times |L| \times |N_d(L)|}$ , we can calculate the target probability  $x_{u,i,l}$ .

### 3.4 TAD-FPMC

FPMC-LR is a successful model for POI recommendation in terms of performance and efficiency; it incorporates sequential data with personalization and locality consideration. However, there is still room for improvement. Li et al. [3] point out that the FPMC-LR approach simply captures the consecutive ordering relations in lieu of considering complex user behavior over time. Take college students, for example. Students have morning classes, afternoon classes, or both in one day. When they have morning classes, some of them have breakfast at a brunch cafe after class. When they have afternoon classes, some of them go for a drink at a bar after class. Simply combining them as one sequential pattern is clearly not sufficient (e.g. recommending a bar after an 8:00AM class). In other words, just recognizing sequential data is not enough for capturing the differences in regular periodic pattern.

To make the system aware of users' time-varying behavioral trends, Li et al. [3] suggest the time-aware FPMC (TA-FPMC) model, which is an improvement over the FPMC-LR model. Both FPMC and FPMC-LR use a third order tensor to construct personalized Markov chains. TA-FPMC takes a similar approach but one step forward. Instead of a third order tensor of user-location-location, it forms a fourth order

tensor user-time-location-location by adding a time factor. Our transition tensor looks like this:

$$\chi \in [0, 1]^{|U| \times |T| \times |L| \times |L|} \quad (6)$$

Using the tensor factorization technique PITF that we used earlier for FPMC and FPMC-LR, we get one rank-3 core tensor and six matrices which capture the following interactions similar to equation 3.

- A: Interaction between the user features  $v_u^{U,T}$  and the time period  $v_t^{T,U}$ .
- B: Interaction between the user features  $v_u^{U,I}$  and the current location  $v_i^{I,U}$ .
- C: Interaction between the user features  $v_u^{U,L}$  and the next location  $v_l^{L,U}$ .
- D: Interaction between the time period  $v_t^{T,I}$  and the current location  $v_i^{I,T}$ .
- E: Interaction between the time period  $v_t^{T,L}$  and the next location  $v_l^{L,T}$ .
- F: Interaction between the current location  $v_i^{I,L}$  and the next location  $v_l^{L,I}$ .

By adding up the corresponding entries in interaction matrices, we get our target value  $x_{u,t,i,l}$

$$\hat{x}_{u,t,i,l} := \underbrace{v_u^{U,T} \cdot v_t^{T,U}}_A + \underbrace{v_u^{U,I} \cdot v_i^{I,U}}_B + \underbrace{v_u^{U,L} \cdot v_l^{L,U}}_C + \underbrace{v_t^{T,I} \cdot v_i^{I,T}}_D + \underbrace{v_t^{T,L} \cdot v_l^{L,T}}_E + \underbrace{v_i^{I,L} \cdot v_l^{L,I}}_F \quad (7)$$

There is another statistical characteristic of LBSN data that has not gotten much attention, the time gap between two successive check-ins. The LBSN data shows that many of two successive check-ins span a large time gap like a year. Based on the assumption that these successive check-ins with large time gaps do not tell much, Li et al. [3] introduce another factor  $D(\Delta t)$  to take into account the time-decaying effect. The final model with all the considerations is called TAD-FPMC. There are a few variations of TAD-FPMC that are highly optimized for certain data, but we focus on the most basic model of TAD-FPMC for simplicity. For more information about TAD-FPMC, refer to [3].

## 4. EXPERIMENTS

In this section, we will address the following questions: 1) How do different POI recommendation models perform in real LBSN data? 2) Why are distance constraint and temporal information particularly important in the POI recommendation?

### 4.1 Datasets

The evaluation used data collected from *Foursquare*, an LBSN that provides POI recommendations. The data includes users' check-in data for New York City from Jan. 2010 to June 2011. The statistics of data is summarized in table 1. [3]

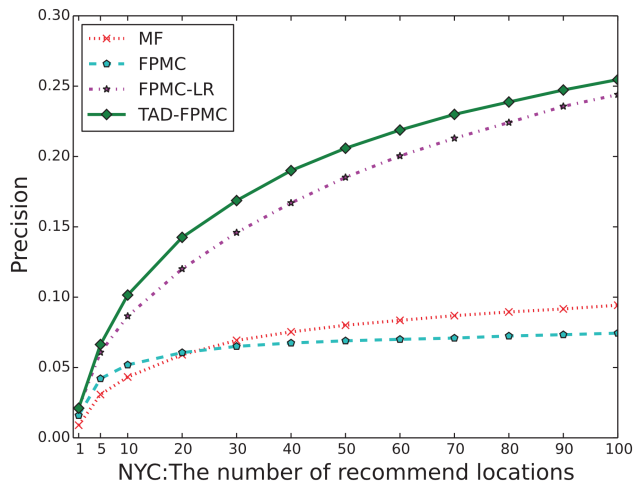


Figure 6: Comparison of location prediction in NYC. [3]

Table 1: The statistics of data. [3]

City	User	Location	Category
New York City	2,581	206,416	249

## 4.2 Evaluation Metric

The main task for the algorithms is providing a list of the top- $N$  recommended locations ordered from the highest probability of user visit to the lowest. Evaluation metrics take advantage of historic data. A location recommendation is considered correct if the user had visited that place in the past. The evaluation metric  $P@N$  (Precision) is defined as:

$$P@N = \frac{\text{the counts of correct predictions}}{\text{the total number of recommendation rounds}} \quad (8)$$

The number of recommendation rounds is a constant representing the number of times a top- $N$  list was generated and checked if correct or not.

## 4.3 Comparison

Figure 6 shows the performance comparison among different algorithms of POI recommendation. The x-axis represents the number of recommended locations,  $N$ , and the y-axis is the precision of the recommendation  $P@N$ . In the graph, there is a huge gap between generic models (MF and FPMC) and POI specific models (FPMC-LR and TAD-FPMC). This verifies the importance of distance constraint and temporal information in POI recommendations. Also, TAD-FPMC slightly outperforms FPMC-LR. More sophisticated variations of TAD-FPMC outperform all the models examined in this paper. The curious reader is referred to [3]. The experiments were run on a quad-core machine with a Core i7-6700K 4.0GHz 8 hyper-threading and memory size of 32GB. [3]

## 5. CONCLUSION

In this paper, we have considered the point-of-interest (POI) recommendation in LBSNs and introduced different approaches for accurate predictions. Factorized personalized Markov chain (FPMC) model is an elegant method com-

binning advantages of Markov chains and matrix factorization, but it was not designed to incorporate distinct characteristics of LBSN data. FPMC-LR is an enhancement of FPMC by only considering nearby locations when calculating POI predictions. TAD-FPMC is a novel POI recommendation system approach that captures complex user behavior over time in addition to inheriting strengths of FPMC and FPMC-LR. From experimental results conducted on *Foursquare* data for NYC, we conclude POI recommendation specific approaches (FPMC-LR, TAD-FPMC) far outperform generic recommendation models such as FPMC. Additionally, TAD-FPMC produces better predictions than any other methods including FPMC-LR.

## Acknowledgments

I thank my senior seminar professors, Peter Dolan and Kristin Lamberty, for their helpful advice and comments throughout the research and writing process.

## 6. REFERENCES

- [1] C. Cheng, H. Yang, M. R. Lyu, and I. King. Where you like to go next: Successive point-of-interest recommendation. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, IJCAI '13*, pages 2605–2611. AAAI Press, 2013.
- [2] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug. 2009.
- [3] X. Li, M. Jiang, H. Hong, and L. Liao. A time-aware personalized point-of-interest recommendation via high-order tensor factorization. *ACM Trans. Inf. Syst.*, 35(4):31:1–31:23, June 2017.
- [4] S. Rendle and C. Freudenthaler. Improving pairwise learning for item recommendation from implicit feedback. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining, WSDM '14*, pages 273–282, New York, NY, USA, 2014. ACM.
- [5] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI '09*, pages 452–461, Arlington, Virginia, United States, 2009. AUAI Press.
- [6] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, pages 811–820, New York, NY, USA, 2010. ACM.
- [7] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM '10*, pages 81–90, New York, NY, USA, 2010. ACM.