Spatio-Temporal Prediction of Location Ratings

Guolei Yang[1] and Andreas Züfle[2]

[1] Iowa State University, Ames IA, USA
yanggl@iastate.edu
[2] George Mason University, Fairfax, VA, USA
azufle@gmu.edu

Abstract. Pint-of-Interest (PoI) rating is a “score" that measures the overall quality of the PoI’s service. As more and more people relies on Location-Based Service (LBS) to find PoIs of interet, PoI rating becomes a crucial factor that affects user’s decision and more. It also determine the rank of an PoI in user’s searching results and location recommendation results. As such, a rating that accurately reflects the quality of the PoI is a fundamental requirement for these applications to be possible. However, PoI ratings generated by LBS is known to be vulnerable to shilling attacks and may not be subjective. This is due to the fact that PoI ratings largely depend on explicit user ratings and written reviews. In this paper, we propose to leverage visitors’ trajectory to predict the rating of a PoI, which does not suffer from these problems. We propose a visiting-pattern-based PoI rating prediction framework, which is based on our analysis of the Foursquare check-in dataset. The core of the proposed framework is a novel regression model that captures the spatio-temporal characteristic of visits from different users. Experiment shows that our method outperforms simply counting based method, as well as method based on state-of-the-art user-location rating techniques.

1 Introduction

The prevalence of GPS-enabled device such as smart phone leads to the thrift of many Location-Based Services (LBS) including GoogleMap and Foursquare, where users can access the information of millions of Point-of-Interests (PoIs) like restaurants and shopping centrers. More and more customers relies on such LBS to search for PoIs of their interest on a daily basis. An important type of PoI information users are looking for on such LBS is **PoI rating**, i.e., a “score" that measures how good the PoI is in terms of service provided. For example, on GoogleMap, each PoI is rated on a scale of 1 to 5 “stars". The more stars a PoI has, the higher it is rated, reflecting that the PoI offers services of high quality.

PoI rating plays a crucial role in customers’ decision making process. Obviously, a user is more likely to use the service of PoIs with higher ratings, comparing with similar (in terms of location and price), but lower-rated PoIs. PoI rating is also an important factor that affects the rank of a PoI in the search result return to a user. Additonally, it is used to determine whether a PoI is to be recommended to users by many recommendation systems. For the above reasons, PoI rating is crucial to the owner a the PoI, and this rating also provides important insights on how to improve his business in order to attract more customers.

A fundamental requirement for PoI rating is it should accurately reflect the quality of service of the PoI. On LBS, PoI rating is usually automatically generated via a data mining process, which takes into consideration of several factors. Such factors usually include individual user's explicit rating and review of the PoI, location and category of the PoI, etc. (e.g.,). This rating generation method is similar to the product rating system for online shopping websites such as eBay and Amazon, which is known to be suffering from several problems, such as:

1. A major concern of this system is it is vulnerable to **shilling** attacks. Shilling attack refers to the behaviour of giving fake ratings and reviews to products, in order to manipulate their ratings. Its effectiveness has been shown by several studies on user-rating based product rating systems. Due to the aforementioned reasons, PoI owners strive to improve their ratings and some of them may resort to this attack. It is estimated that -% of user rating on ... are likely to be fake. Since anyone can rate PoIs on LBS such as GoogleMaps, there is no reason to believe that their PoI rating systems are immune from such attacks.

2. This system relies on complex text-mining algorithms. Written review is usually more trustworthy and provides more information about a product comparing with simple a rating score. However, extract useful rating information (e.g., positive/negative labels) from written reviews is known to be a challenging task. Although a few methods has been proposed for automatic review-text mining, the mining process is still expensive and the accuracy is not very satisfactory. More importantly, when used for PoI rating generating, the errors from multiple user's review will accumulate.

3. The system requires extra user efforts and may not be objective. It require users to explicitly give their ratings to a location and/or write reviews, which requires extra user efforts and may not be objective (e.g., a “score" can be interpreted very differently by different users, inaccurate description, only users who strongly dislike or like the PoI are motivated to rate or write reviews, etc.).

The root of the above problem is that this method strongly relies on individual user's explicit ratings and written reviews, but their actual behaviours. However, a unique difference between a PoI and a normal product is that for a PoI, an LBS can not only know who visited it, but also knows the trajectories of these visitors. This is made possible by either keep tracking user's locations, or ask users to voluntarily report check-in to the PoI. For example, the publicly available Foursquare dataset contains more than 3 millions of user self-reported check-ins.

These trajectories provides a unique dimension that could be leverage for PoI rating, which provides a possible solution to the aforementioned problems of normal product rating method. To this end, we study the spatio-temporal PoI rating prediction problem. We aim to answer the following question: *Is it possible to accurately predict the rating of a PoI by mining the trajectories of its historical visitors?* We want to point out that our problem is different from user-location rating prediction, where the goal is to predict whether *a specific user* likes a PoI, but not to predict the overall rating of a PoI. The former finds its applications in making personalized recommendations, while the later is crucial for the aforementioned applications.

There are several potential advantages of using trajectory data for PoI rating: (i) It is prone to alteration by fake-user-ratings and spam/bot-user-ratings. It is much harder and more expensive to forge a valid trajectory than to submit a fake rating score or review. As such, it is resistant to common shilling attack. ii It does not require an intermediate process to extract label information from written reviews. (iii) no extra user effort to capture their opinions such as filling in rating forms or submit a written review, and (iv) it is based on user's behaviour, thus could be more objective.

A straightforward way to predict the rating of a PoI is to simply count the number of visitors. This is based on the assumption that high-rating PoIs are usually popular locations since they are more attractive to costumers. Although the assumption is reasonable to some extend, it has some obvious drawbacks. First, a PoI being popular could only be the result of its location. For example, a Starbucks coffee shop located in a busy airport could have much more visitors than another Starbucks in suburb area, despite that the products they serve is largely the same. Second, user's choice of PoIs may be affected by time. For example, a customer who is on the way to work is likely to choose a restaurant for breakfast only because it is close to work and serves food fast, not because the food is exceptional. Third, the visits from different user may different predictive power since some user may be more selective in choosing PoIs in certain circumstance. For example, a user who is more familiar with an area may knows better which restaurant have the best service.

To address the above challenges, we propose a *visiting-pattern-based PoI rating prediction framework* that captures the spatio-temporal characteristic of visits. The visiting-pattern is extracted from a user's trajectory, which reflects how often the user visits a PoI, the visiting time, and the travel distance. The proposed framework is based on a comprehensive analysis of the intuitive relation between user visits to a PoI and its rating over the Foursquare dataset. The core the proposed framework is a novel regression model that takes into consideration the difference between visitors, visiting time, PoI locations, etc. We show in experiments on real world dataset that the proposed method outperforms not only the simple counting based method, but also the method based on state-of-the-art user-location rating techniques. We summarize our contributions as follows:

- We formally define the spatio-temporal PoI rating problem, and conduct the first systematic study of the problem.
- We perform a comprehensive analysis of the Foursquare check-in dataset, in order to identify relations between user visits and PoI ratings.
- Based on our analysis, we propose a visiting-pattern-based PoI rating prediction framework. It captures the spatio-temporal characteristic of visits by extracting patterns from user's trajectories. The core the proposed framework is a novel regression model that takes into consideration the difference between visitors, visiting time, PoI locations, etc.
- The proposed technique is implemented and evaluated on real world dataset. The results show it outperforms not only the simple counting based method, but also the method based on state-of-the-art user-location rating techniques.

The rest of this work is organized as follows. We survey the state-of-the-art on location rating in Section **??**. Then, we formally define the problem spatio-temporal PoI rating in Section **??**. Section present our analysis of the Foursquare check-in dataset.

Details of the proposed framework is described in Section **??**. Our solution is evaluated in Section **??**. And finally, we conclude our work in Section **??**.

2 Related Work

The problem studied in this paper is related to user-location rating prediction, which is a major research issue in location recommendation systems. Unlike the PoI rating discussed in this paper, user-location rating is a rating a specific user gives to a location to reflect his personal feeling towards the location.

In general, there are two types techniques of user-location rating. 1) Classification-based techniques. It takes a user-location pair as input and label it with one of two classes: “like" or “dislike". Non-binary classifier can also be built model the case where users have more than two rating options. But most commonly, the interest is only to determine if the user will like the location or not, not to predict the “degree" of user's preference of the PoI. 2) Ranking-based techniques. The input is a user and a list of candidate locations, and the goal is to rank the candidate locations by the probability that the user likes the location. Ranking-based technique is also referred to as check-in prediction in some literature, to emphasis that the goal is to predict which location(s) the user is most likely to visit next given his current trajectory.

General user-product rating prediction techniques can be directly used for user-location rating by simply treating locations as products. One of the most widely used technique is Matrix Factorization, which makes predictions by mapping both users and locations into a set of latent features. The features are leant from known user-location ratings in the training dataset. However, generic techniques such as Matrix Factorization do not take into consideration spatio-temporal characteristic of locations or user movements. A series of spatio-temporal user-location rating prediction techniques have then been proposed to improve the prediction performance by leverage spatio-temporal data.

It is worth mentioning that user-location rating prediction can also be used to address our problem if combined with traditional user-rating based method: Since the overall PoI rating is usually derived from individual ratings and reviews of the PoI, we can first predict the individual ratings of the PoI based on the spatio-temporal information of its visitors, and then generate the PoI rating based on the predicted individual user-location ratings (e.g., take the average visitor rating as overall PoI rating like some LBS). However, the prediction error introduced in the user-location rating prediction process will accumulate in the final prediction results. This is supported by our experiment, which shows even the state-of-the-art user-location rating prediction technique cannot provide good accuracy for PoI rating comparing with our technique as a result of this error accumulation.

3 Problem Definition

In this section we formally define a user-trajectory, and define our notion of a user-site-stay, which we are going to use to extract features to estimate the affinity between a user and a site later in Section 5. We first start by defining a trajectory as follows.

Definition 1 (Spatio-Temporal Database). *Let $\mathcal{U}$ denote a set of unique user identifiers, let $\mathcal{G} = [-90, 90] \times [-180, 180]$ denote the space of longitude/latitude geo-coordinates, and the $\mathcal{T}$ denote the time domain. A* spatio-temporal database $\mathcal{D} \subseteq \mathcal{U} \times \mathcal{G} \times \mathcal{T}$ *is a collection of triples $(id \in \mathcal{U}, (lat, long) \in \mathcal{G}, t \in \mathcal{T})$. Each triple $(u, s, t) \in \mathcal{D}$ is called an observation.*

We group a spatio-temporal database into observations of the same user, denoted as user-trajectory, formally:

Definition 2 (User-Trajectory). *Let $\mathcal{D}$ be a spatio-temporal database and let $u \in \mathcal{U}$ be a user. The set*

$$\mathcal{D}(u) := \{(u', (lat, long), t) \in \mathcal{D} | u = u'\}$$

is called the user-trajectory of user u.

In order to obtain recommendation information from a user-trajectory, we need to link the user-trajectory to sites, such as restaurants and hotels. For this purpose, we join a spatio-temporal database with a database of points of interest (such as provided by Open-Street Map) like restaurants and hotels. Next, we define our notion of a *stay*. A stay is an event of user visiting a site, enriched by the duration of the stay.

Definition 3 (Stay Trajectory). *Let $\mathcal{D}$ be a spatio-temporal database and let $\mathcal{S} \subseteq \mathcal{G}$ be a collection of $(lat, long)$ pairs of sites. A stay is a triple $(u \in U, s \in \mathcal{S}, (t_{start}, t_{end}) \in T \times T)$, indicating that user u has stayed at site s from time t_{start} to time t_{end}. We let $(\mathcal{D} \bowtie \mathcal{S})$ denote the set of all stays mined from all trajectories $\mathcal{D}$ using all sites in $\mathcal{S}$. The sequence of all stays a user $u \in \mathcal{U}$ is called the stay-trajectory $(\mathcal{D} \bowtie \mathcal{S})(u)$ of u, defined as:*

$$(\mathcal{D} \bowtie \mathcal{S})(u) := \{(u, p, (t_{start}, t_{end})) \in \mathcal{S} | u = u'\}$$

Finding stay points in trajectory and PoI databases is a research topic that has raised attention in the past. In this work, we make the explicit assumption that stay points of a trajectory are already given, as we use trajectory datasets where stay points are explicitly labeled. We discuss this assumption and related work on stay-point detection in Section **??**.

Given stay-trajectories for each user, the challenge of this work is to predict the rating between users and a site. As site is PoI which can be rated by a user, such as a restaurant or a hotel. We assume that we have a recommendation database, where users can rate sites. We assume normalized rating values in the interval $[0, 1]$, where 0 corresponds to the lowest rating and 1 corresponds to the highest rating.

Definition 4 (User-Site Recommendation Database). *A recommendation database $\mathcal{R}$ is a set of user ratings $\mathcal{R} \subseteq \mathcal{U} \times \mathcal{S} \times [0, 1]$.*

The task of this work is to predict the triples in R. That is, the challenge is to predict the rating that a user $u \in \mathcal{U}$ will give to a site $s \in \mathcal{S}$, using a spatio-temporal given in $\mathcal{D}$.

4 Data Analysis

The first step of our user-site-recommendation approach requires to map a raw trajectory $\mathcal{D}(u)$ of a user u to a stay-trajectory $(\mathcal{D} \bowtie \mathcal{S})(u)$. Such stay points can be detected by using existing work such as proposed in [4, 8, 7, 5]. All of these works use a distance threshold θ_d and defines a stay as the duration of time where the trajectory does not exceed a distance of θ_d to a PoI. In this work, we entirely circumvent the step of implicit stay point detection, by using data that has explicit stay points. Therefore, in our experimental evaluation we use Check-in data from location based social networks (LBSNs), which explicitly contain the stay points of users. Clearly, by circumventing the problem of stay point detection, we limit our experimental evaluation to LBSN Check-in data, which is relatively small (hundreds of megabytes of data), compared to large raw trajectory databases (Terrabytes of data). Yet, we postulate that our relatively small FourSquare Check-in dataset, allows to effectively predict the user rating of a restaurant. This hypothesis is also supported by our experiments.

5 Proposed Method

Our goal is to extract a set of spatial-temporal features from a user's trajectory, which can be directly used to predict the rating of a PoI. There are many features that could potentially be related to a user's feeling of a PoI. We discuss in this section a set of common features we select for the rating prediction problem, and the methods we use to extra such features. It is easy to understand why these features are selected since they are intuitive.

5.1 Extracting the frequency of visits of a user

Our assumption is, if a user repeatedly visits a PoI, e.g., always dine in the same restaurant, it strongly suggests that the user favours the PoI. On the other hand, if a user visits a PoI only once and never comes back, it suggests the user dislike the place. Therefore we choose the frequency of visits as a feature. Here, each stay-point is considered a visit.

In order to extra frequency information from a user's trajectories, we count for how many time the user visits a PoI in a given time period. Each check-in to the PoI is counted as one visit. Users are then assigned into several frequency groups (Table 1) based on how often they visit the PoI. Note that if a user has never visited a PoI, he cannot rate it. Thus we do not assign any group for such users.

We note a user's behaviour may vary over time. For example, if a user likes a shopping center, he may visit a shopping center very frequently during Thanksgiving and similar holidays, while visit the same place only once in several month during the rest of the year. As a result, the timing window used to count visits can have significant impact on the user's group assignment. For the same user, if we consider only visits happened in the Thanksgiving week, the user belongs to the "At least one visit per week " group. However, if we look at the year-long visits, he may be grouped into "Less

Table 1: Groups based on how often user visits a location

ID	**Frequency groups**
1	At least one visit per day
2	At least one visit per week
3	At least one visit per month
4	Less than one visit per month

than one visit per month" since his visits are averaged over twelve months. We propose a simple way to mitigate this problem.

First, we use timing windows with different sizes simultaneously to compute the frequency of visits of a user. Specifically, given a user's trajectory data for a period of n days, for each week/2 weeks/month within these n days, we count the number of visits that falls in the same week/2 weeks/month. We then use this count to compute the user's group of that specific period. Here we use week and month as nature timing windows, but in practice the size of timing window can be arbitrary. Second, when a user's visit frequency demonstrates inconsistency in different time period with the same timing window size, we use his maximal visiting frequency, e.g., if a user visits a location 3 times a week in week 1, but 0 times in week 2, the user is then categorized as "At least one visit per week" in this case.

Additionally, for each PoI, we calculate three numerical features that also describe a user's visiting pattern to the PoI: 1) **Average duration between two consecutive visits of the user**, 2) **Minimal duration between two consecutive visits of the user**, and 3) **Maximal duration between two consecutive visits of the user**. These information are supplemental to the frequency groups.

5.2 Extracting the length of stays of a user

Given a trajectory of a user, the length of stays can be inferred by examining the interval between consecutive check-ins of the user. If two visits are reported within a time threshold τ, e.g., 30-minutes, it is safe to consider the two check-ins belong to the same stay. Thus, the minimal length of this stay is the time frame between the two check-ins. Similarly, an upper bound of the length of stay is the time between a check-in to the location and the closest check-in to a different location. This provides us with a coarse way to estimate the average length of stay of users to a PoI. This method is most accurate if the user's location is reported periodically with small intervals, or whenever the user changes location.

5.3 Extracting the travel distance of a user from its home base

If a user is willing to travel a long distance from his home/work area to a PoI, it is likely that the PoI is attractive to him. Unfortunately, measuring the travel distance from one's home/work can be challenging. This is because users' home or work address or coordinates are usually not explicitly marked in a spatial-temporal dataset due to privacy concerns.

We use distance-based location clustering (e.g., weighted k-means [2], where the weight of a location is the number of visits by the user) to estimate a user's home base, i.e., an area where he is likely to live/work. Locations a user visits frequently is likely to formulate a dense cluster in the area where he lives and works [1, **?**]. If there exists a location that is isolated from any of these clusters, we deem such a location as "far from home". It requires extra effort for the user to travel this location. Willing to make such effort indicates the user may like the place. An example of home base and isolated locations is illustrated in Figure 1.

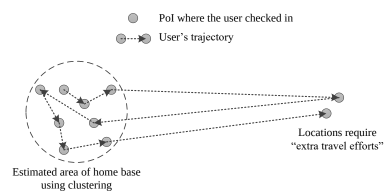

Fig. 1: Example of home base and isolated locations

For each isolated location, we estimate the travelling effort for the user to reach the location. Using the distance between the isolated location and the user's home base may be biased. A user who likes to drive can easily travel to a mall several miles away from home, but for someone who does not have a car, travelling the same distance means significantly more effort. Instead, we use the relative travelling distance. For a location l, we compute $D_l = d_l/\overline{d_{home}}$ where d_l is the minimal travel distance between l and any home base location, and $\overline{d_{home}}$ the average travel distance between locations within the user's home base.

5.4 Extracting the type of PoI

A user's spatial-temporal behaviour can be very different in different types of PoIs. Suppose a user gives high rate for both a coffee stand and a theatre. It is common for a user to visit the coffee stand every day or even multiple times a day, while such a visiting pattern is not likely to appear for his visiting to the theatre. Given the diversity of locations, we believe the type of different PoIs serves as an important feature in the rating prediction process.

Fortunately, in many spatial-temporal datasets (e.g., [6]), a category is given for each PoI. Nevertheless, a dataset may provide only coordinate of visited locations with no additional information. One way to infer the type of PoI in this case is to match the coordinates to PoIs using geo-info systems such as GoogleMap, which provides detailed category information of PoIs.

5.5 Discussion

Due to limited space, we briefly discuss some other useful features that we conjecture are related to users' rating of PoIs.

Regional difference: User's visiting pattern to some PoIs can be region-sensitive. Consider an example of two restaurants. Restaurant A locates near a busy train station while restaurant B is in suburb region. Even if the B provides better service than A, A may attract much more visitors comparing with B due to its better location. A possible way to mitigate the problem is to add a feature that describes region of a location (e.g., "Downtown region", "Less popular region", etc.) but it requires extra geo-demographic information which cloud be hard to collect. A simpler and efficient way is to explore the relation between nearby PoIs of the same type. The reason is intuitive. For example, given several restaurants close to each other, if all of them are visited by 100 users per day, except for one restaurants which has only 10 visitors per day, it is very likely that this restaurants has a bad rating.

Scale of the PoI: It is worth mentioning that the scale of a PoI has an obvious impact on the total number of visits to it. A small convenient store, regardless of how users like it, is not likely to have even close number of visitors comparing with a large super market. However, to our knowledge, these is no convincing way that can accurately estimate scale of a PoI. We leave this for future exploration when such data is available.

Social connection between users: The prevalence of LBSNs makes it easy to learn social connects between users. If several users who are socially connected check in at a PoI at the same time, it is likely that they have a group social event at the PoI. The fact that a group of friends choose to meet at a PoI suggests most of the users in the group like the place. We may also assume a user's rating of some PoIs is affected by the rating of his friends, given that they frequently visit a similar set of PoIs. In general Group-visits may demonstrate different pattern comparing with individual visits, and therefore can be used as a distinguished feature.

6 Experimental Evaluation

6.1 Experiment Settings

We evaluate the proposed technique over the FourSquare check-in dataset [6]. The dataset contains 227,428 check-ins in New York city and 573,703 check-ins in Tokyo collected in 10 month. A PoI type is given for each record. We use the user ratings on FourSquare as ground truth. Specifically, FourSquare uses a 10-point rating system, but a user cannot give arbitrary points to a PoI. A user can select from three options: "Like" (count as 10 points), "Neither like nor dislike" (count as 5 points), or "Dislike" (count as 0 point). Which PoI is liked by a user is publicly available. We have implemented f schemes:

- Linear regression (LR) This scheme predicts user ratings with a model generated by simple linear regression with the proposed features.
- Matrix factorization (MF) The simple non-negative matrix factorization [3] for recommender systems. We use matrix factorization to predict the rating of each users

who have visited a PoI, and then use their average rating score as the "overall" rating of the PoI.
- Deep neural network (DNN) The scheme predicts user ratings with a model trained by deep neural network as proposed in Section **??**, using the proposed features.

We have also implemented two simple baseline approaches for comparison:
- Random We randomly generate a score in $[0, 10]$ as user rating for each PoI.
- Average We use the average rating of all the PoIs in the tested dataset as the prediction value.

For LR and DNN, we use the four types of features discussed in Section 5. More parameters of our experiment is given in Table 2. Our experiment platform is a virtual machine with Intel Xeon 64-bit 8-core CPU running on 2.93GHz and 32GB RAM. The algorithms are implemented with Microsoft Azure machine learning toolkit and modules.

Table 2: Groups based on how often user visits a location

Parameter	**Default value**
Number of users	5000
Number of PoIs	1500
Training set size (%)	10%
Cost function	Mean Square Error
Neural network depth (for DNN)	4
Neurons per layer (for DNN)	12
Output function (for DNN)	sigmoid
Number of latent features (for MF)	8

6.2 Data Cleaning

A few data cleaning steps are needed to prepare the dataset. First, we remove the *inactive* users from the dataset. We define inactive users as those who have less than 1 check-ins per week in the 10 month period, and have less than 10 liked locations. The lack of data makes it hard to extra features for these users. Second, we remove the PoIs which were not rated by any of these users, since the ground truth is not available for such PoI. After the cleaning process, we extract the check-in information of approximately 5000 most active users and 1500 PoIs they have visited from the dataset. We show this small set of users/PoI is sufficient to demonstrate the effectiveness of the proposed rating prediction method.

In order for MF to work, we also generate a user-PoI rating matrix based on the users and PoIs appeared in the FourSquare dataset. The user ratings are directly crawled from FourSquare.

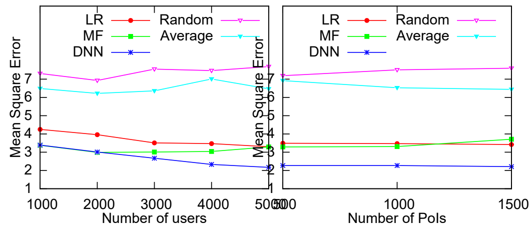

Fig. 2: MSE of different prediction schemes

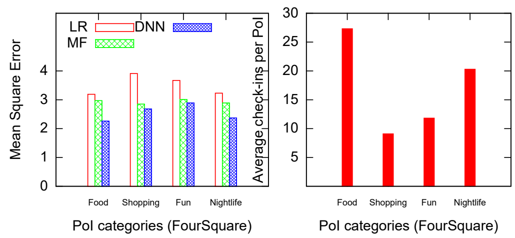

Fig. 3: MSE of different PoI categories

6.3 Prediction Results

We compare the Mean Square Error (MSE) of rating prediction results of the three schemes. We adjust the number of users and PoIs, respectively, and exam the performance trend of the compared schemes. The result is plotted in Figure 2. The proposed DNN prediction method demonstrate clear advantage over the other schemes for a moderate number of users. The performance of LR is also close to that of pure matrix factorization for larger number of users. The performance of DNN increases as the number of users increases. This implies a larger group of users overall have more reliable spatial-temporal features. In contrast, the performance of MF drops as the number of users/PoI increases. As the size of user-PoI rating matrix increases, it becomes more spares, making it harder for MF to generate accurate predicts.

We also shows the MSE of the prediction result for PoIs in four different categories, namely “Food", “Shopping", “Fun", and “Nightlife". These PoI categories are provided by FourSquare. There appears to be a difference of prediction accuracy between PoI categories (Figure 3-a). For the proposed methods, the prediction result for “Food" and “Nightlife" shows a lower MSE comparing with that of “Shopping" and “Fun". A closer look at check-in data in different PoI category reveals that users reported less check-ins for “Shopping" and “Fun" PoIs comparing with that of ‘Food" and “Nightlife" (Figure 3-b). Thus the relatively worse performance can be explained by the lack of data, which can affect the quality of extracted features.

Figure 4 compares PoIs with different ratings in terms of the (normalized) average number of user in each frequency group, as showed in Table 1. We define high rating PoI as a PoI with a overall score of 7.5 or higher out of 10. While a low rating PoI is one with an overall score of 2.5 or less. In general, highly rated PoIs have more

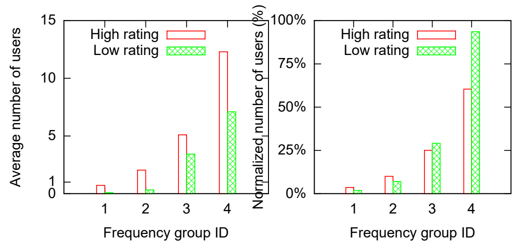

Fig. 4: User frequency group distribution

visitors. Furthermore, we observe that a highly rated PoI attracts more frequent-visitors among all the visitors, comparing with PoIs with low ratings. These results support the assumptions we made in Section 5.

7 Conclusions

Location-based social networks such as FourSquare, Yelp, TripAdvisor and OpenRice allow users to "check-in" and rate sites such as restaurants and hotels, to quantify their experience with that site. However, the vast majority of users does not use such online platforms to publish their opinion. To improve such user-site recommendation systems, we propose to exploit spatio-temporal data to estimate user-site ratings of any user. Therefore, we analysed trajectory data to find discriminative features indicating user preferences, such as their frequency of visit of a site, duration of visit, and distance from their home base. Using these features, we use existing (explicit) user-site recommendations to learn the relationship between our proposed features and the this ground-truth. Our experiments show, that our solution allows to drastically reduce the user-site-rating prediction error by exploiting spatio-temporal data. To leverage our solution to large spatio-temporal datasets, our next step is to automatically detect stay-points, rather than using pre-labelled check-in data.

References

1. P. Golle and K. Partridge. On the anonymity of home/work location pairs. In *PerCom*, pages 390–397. Springer, 2009.
2. K. Kerdprasop, N. Kerdprasop, and P. Sattayatham. Weighted k-means for density-biased clustering. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 488–497. Springer, 2005.
3. Y. Koren, R. Bell, C. Volinsky, et al. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
4. Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma. Mining user similarity based on location history. In *ACM SIGSPATIAL*, 2008.
5. X. Xiao, Y. Zheng, Q. Luo, and X. Xie. Finding similar users using category-based location history. In *ACM SIGSPATIAL*, pages 442–445, 2010.
6. D. Yang, D. Zhang, V. W. Zheng, and Z. Yu. Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1):129–142, 2015.

7. Y. Zheng, X. Xie, and W.-Y. Ma. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.*, 33(2):32–39, 2010.
8. Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from gps trajectories. In *WWW*, pages 791–800. ACM, 2009.