Foursquare NYC Check-In Visualization and Recommendation Engine

Neel Kapadia
Department of Computer Science, NCSU
Raleigh, US
ntkapadi@ncsu.edu

Tushita Roychaudhury
Department of Computer Science, NCSU
Raleigh, US
troycha@ncsu.edu

Sanya Kathuria
Department of Computer Science, NCSU
Raleigh, US
skathur2@ncsu.edu

Abstract - **As the surge for Location Based Social Networks has grown, it has become easier to obtain the activity data for hundreds of millions of user-driven footprints (that is "check ins"). This information can be used for studying social and temporal characteristics of how people use these services and to model patterns of human mobility are significant for improving future mobile and location-based services. We investigated the Foursquare check-ins dataset containing 227,428 check-ins in New York city and report a quantitative assessment of human mobility patterns by analyzing the spatial and temporal aspects associated with these footprints. Each check-in is associated with a time stamp, its GPS coordinates and its semantic meaning which is represented by their venue-categories. In this paper, we attempt to explore the behavior of Foursquare users in New York city and then visualize the dataset to find interesting patterns. We then present a recommender system which suggests a particular set of venues (such as restaurants) to a user and then further also develop a second model for recommending a set of users to a venue, keeping in mind both user-preferences, users' location and popularity of the venues.)**

Keywords- recommendation; venue; foursquare; spatial visualization

I. INTRODUCTION

As the growing ubiquity of GPS-equipped smartphones, Location Based Social Networks (LSBNs) have gained increasing popularity in recent years. Foursquare is such a location-based social networking application for mobile devices. This service is available to users with GPS enabled mobile devices, such as iPhones and Android. In such location based social networks, a user activity is mainly represented by "check-in" which indicates that a user visited a POI at a certain time. These large-scale users generated digital footprints give an opportunity to understand the spatial and temporal features of user activity.

In terms of scale, the Foursquare service alone claims over 6 million registered users (Foursquare 2011) and nearly 1 million check-ins per day (Grove 2010). By mining and visualizing these activity records based on various parameters like check-ins through the day, peaks for venue categories and scatter plot for data points, we are able to understand user spatial temporal activity preference which can then enable various location-based applications.

In this paper, we have attempted to make the following contributions:

- Analysis of New York Foursquare data
- Recommending places to a user
- Recommending users to a place

A. Analysis of New York Foursquare data

In our first module, we have done explanatory and exploratory visualization through the lens of Foursquare user check-ins dataset to infer insights about New Yorkers. Based on parameters like venue categories, hourly check-ins, daily check-ins and check-ins count, we used different kinds of maps to look at users' behaviors through many different timeframes of the year for detecting patterns.

After analysis of the dataset, we attempt to answer the following questions: "Which venue is a mobile user most likely to visit given his/her current context, including time and location?" and "Given a venue (e.g. an Indian restaurant), select individuals who are likely to visit it?". For example, a user, John, is having Indian cuisine at ⌈ 34.1211, -80.2281⌉ at 14:21 on Wednesday. Next time, knowing John is interested in going to an Indian restaurant on Friday, a recommendation of Indian food in a nearby POI would be persuasive.

B. Recommending Places to Users.

In our second module, we address the solution for the first question with a POI recommendation system to suggest places to users by mining users' activity records from which we are able to understand user spatial temporal activity preference which can then enable us to make an informed suggestion

C. Recommending Users to Places.

In our third module, we address the solution for the second question with a location-based recommendation system to suggest users to places. Little research has been done in this area and it has a lot of potential applications including Targeted advertising, Social marketing, etc. Our third module, location-based visitor recommendation system is based on the idea to use users' data and recommend individuals to venues.

This paper is organized as follows: We present a high-level overview of related work in the next section. In Section 3 and 4, we describe the data set characteristics and implementation methodologies that we use in our study. In Section 5, we present the analysis and findings. Finally, we present our conclusions and future work.

II. Related Work

The role of geography and location in online social networks has recently gained an increasing interest. We studied the where's and when's of over 2 million check-ins across the New York city. It helped us to learn about human mobility patterns revealed by these check-ins and explore various parameters that influence human mobility, including the venue category popularity, users' preferences, and geographic constraints. Most of the work has been done around recommendation, prediction, and personalization of venues based on check-in data.

Examples of such work are personalized point-of-interest recommendation (Liu et al. [1], Yuan et al. [2]) and location recommendation by Gao et al. [3]. Facebook researchers analyzed the distance between Facebook users' social relations, and utilized locations of a user's friends' to predict the user's geographical location (Backstrom, Sun, and Marlow 2010). (Cheng, Caverlee, and Lee 2010) modeled the spatial distribution of words in Twitter's user-generated content to predict the user's location. Besides locations, researchers have also explored temporal dynamics associated with online social activities (Golder, Wilkinson, and Huberman 2007). A couple of studies on Foursquare patterns are the work by Cheng et al. [4] and Noulas et al. [5], where they analyze the spatio-temporal characteristics of check-ins.

Since the objective of this paper is to infer user spatial temporal activity preferences in LBSNs, we first briefly survey these research works on user activities from two perspectives: 1) user mobility perspective which focuses on modeling user mobility patterns by leveraging spatial temporal regularities [6]; 2) user preference perspective which usually focuses on inferring venue categories preference based on users' historical check-ins and preferences. We then present the research work by considering POI categories as user activities to recommend places to users and another model to recommend people to venue based on categories choice, geographical constraints and history of their check ins.

III. Methodologies

Module 1: Foursquare Data Visualization

Based on parameters like venue categories, hourly check-ins, daily check-ins and check-ins count, we have used different kinds of maps to look at users' behavior through many different timeframes of the year and specific venue categories for detecting patterns.

Module 2: Recommending places to users

Here, we design a recommendation formula that suggests places to a user. For the user in consideration, we ask for the category of the venue he/she would like to visit and also the timeframe of the visit. The User ID and Category are used as filtering criteria on the dataset.

Once we have the relevant subset of the dataset, we calculate:

p_close: Gives a measure of effect that the distance from a particular venue has on the user's choice of visiting it. Given a user we calculate how many of the user's past check-ins fall within a specified distance threshold from a venue. Haversine distance is used for this calculation. The check ins falling within the threshold are then divided by the total number of user check-ins. This is a measure of how likely a user will visit a particular venue considering the amount of check ins he made around the venue.

p_like: Gives a measure of the venue's popularity. Our assumption here is that more number of check-ins at a particular venue translate to higher popularity of the venue. We calculate the number of check-ins at the venue in consideration and divide it by the total number of check-ins in that venue category. We also apply a penalizing factor based on time.

Weighing the parameters:

Effect of distance (w1): The distance the user has to travel to reach a particular destination has a major determining factor on the user's decision to visit the place. We try to account for this by weighing down the probability of a user visiting a far-off place in case distance is important to a user. We analyze the locations of past user check-ins to determine user behavior.

Effect of time (w2): The idea here is that some venues are visited popularly during some time frames of the day. We considered the difference between current time and average time of all check-ins at that place. The higher the absolute value of the difference, the higher the penalty (lower the w2).

Final formula:

$$p = (w_1 * plike) * (w_2 * pclose)$$

Algorithm:

```
Input: User ID, Category, Time
Output: List of top 20 suggested places
A. Initial calculations:
1. Get check-in count per venue in the category: venue_group
B. Calculating pclose and plike:
1. For each venue v in venue_group:
2.     Calculate distance of every check in of user from v
3.     Select all check ins within a selected threshold t
4.     Calculate p_close for v as → (number of check ins
       within t) / (total number of check-ins by the user)
5. For each venue v in venue_group:
6.     Calculate the number of check in at v (c1)
7.     Calculate the total number of check ins in the
       category of v (c2)
8.     Calculate p_like for v as → (c1) / (c2)
C. Calculating weights w1 and w2
1. For each venue v:
```

```
2.      Calculate the average hour a at which the venue is visited
3.      Calculate the hourly difference between time input t and a
4.      Determine w2 based on the time difference
5. **For** the given user u:
6.      Run spatial clustering algorithm on user check in data to determine the frequently visited locations
7.      **For each** venue v in venue_group:
8.              Calculate the distance of v from the identified clusters to arrive at w2.
```

Module 3: Recommending users to places

We design a formula that recommends users to places. For the venue in consideration, we see which users are most likely to visit that venue.

To start with, we take Venue ID as the input. Once we have the relevant subset of the dataset pertaining to the venue, we calculate:

P_go: Gives the probability that a user will go to any venue at all. We calculate this as number of user check-ins divided by the total number of check ins.

P_close: Gives a measure of the effect the distance from a particular venue has on the user's choice of visiting it. We analyze the average coordinates of user check in and consider it as user's center of mass. This is calculated for each user. Inverse of the distance of this center from the venue coordinates (and raised to a scaling factor α determined by the venue category) is calculated to determine p_close.

P_like: For each user, the venues similar to the ones previously visited by the user. This is used to measure the likelihood of the user visiting the venue.

Final formula:

$$p\ (go|like, close) = \frac{(\,\mathrm{p(like \mid go, close)}\ *\ \mathrm{p(go \mid close)})}{p(like|close)}$$

Algorithm:
Input**:** Venue ID

Output**:** List of users likely to visit the given venue

A. Calculate p_go, p_close, p_like:

```
1. **For each** user u:
2.      Calculate number of check-ins by u/ total number of checkins as p_go
3. **For each** user u:
4.      Calculate the average location of all checkins on u as center of mass c of u
5.      Calculate distance between c and venue location as d
6.      Calculate inverse of d raised to $\alpha$ as p_close of u
7. **For each** user u:
8.      Apply collaborative filtering on a user-by-venue matrix to obtain a rating l for the venue
9.      Calculate p_like as number of venues visited by u with rating l / total number of venues visited
```

B**. Calculate the final formula (B)** for every user and pick the top 20 users with the highest probability. These serve as user recommendations to the given venue.

IV. Implementation

Module 1: Visualizing Foursquare Data

We used the csv Foursquare dataset file and split it into panda data frames to get some smaller insightful parts such as check-ins hourly, check-ins monthly, check-ins category and counts (frequencies) for each of them. Firstly, we plotted the dataset using latitude and longitude for creating an exploratory Google map visualization using gmplot. Secondly, we created visualizations to understand how the number of check-ins vary through the day by plotting histogram for 2 million check-ins, rounded to roughly around 180,000 check-ins venues each month, allowing us to get inferences about users' behavior through many different timeframes of the year that is according to days, monthly and hourly. We have further created line graphs to visualize the number of check-ins throughout the day for six different kinds of venues to see the differences in check-in patterns between different venues categories like bars, subways, coffee shops, college buildings, gym and clothing stores.

Module 2: Recommending places to users

i. **Probabilistic Approach**

We use a probabilistic approach to recommend places to users.

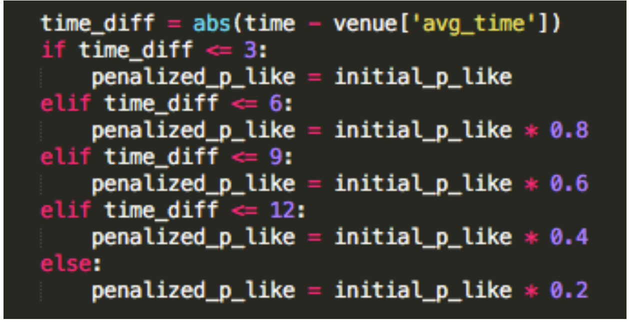
$$p = (w_1 * plike) * (w_2 * pclose)$$

$$plike = \frac{\textit{number of checkins in venue radius}}{\textit{total number of checkins at all venues in that category}}$$

$$pclose = \frac{\textit{sum of number of checkins of top 30\% nearby venues}}{\textit{sum of number of checkins of all venues}}$$

ii. **Considering the "current time" factor**

We also consider the current time at which the user wants suggestions i.e. the algorithm takes care of the fact that a coffee place which is famous for breakfast should not be suggested for evening coffee breaks. The penalty constraint is used to penalize plike based on the difference between current time and the average of all check-ins at that venue as show in in figure 1.



```
time_diff = abs(time - venue['avg_time'])
if time_diff <= 3:
    penalized_p_like = initial_p_like
elif time_diff <= 6:
    penalized_p_like = initial_p_like * 0.8
elif time_diff <= 9:
    penalized_p_like = initial_p_like * 0.6
elif time_diff <= 12:
    penalized_p_like = initial_p_like * 0.4
else:
    penalized_p_like = initial_p_like * 0.2
```

Figure 1

iii. **Haversine Distance**

To calculate pclose, we use a metric called Haversine Distance which is used to measure great circle distances.

iv. **Determination of weights (DBSCAN, Convex Hull)**.

First, we remove the outliers/noise points from the data containing all user check-ins using DBSCAN and then use convex hull to determine the total area covered by the user check-ins. If the area is large then we imply that the user gives more preference to the venue quality than the distance (higher w1 and lower w2) and vice versa.

Module 3: Recommending users to places

i. **Bayesian Probabilistic Modelling**

$$p\,(go|like, close) = \frac{(\,\mathrm{p(like \mid go, close)}\ *\ \mathrm{p(go \mid close)})}{p(like|close)}$$

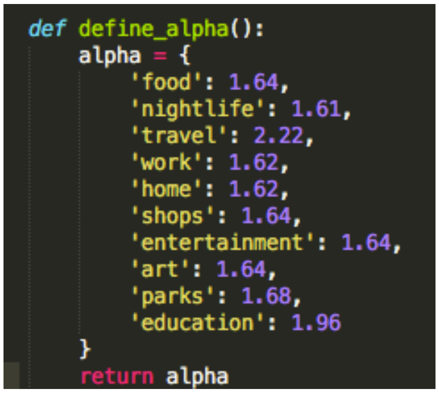
ii. **Calculation of pclose**

$$\mathbf{pclose = (1\ /\ (d_{ui}))^{\alpha}}$$

Here, d_{ui} is the distance of the venue i from user u's center of mass. Center of mass is average of all the check-in locations of that user. α is a constant defining how much the distance matter in a user selecting a venue as shown in the figure 2.

iii. **Center of Mass**

The users center of mass is defined as the mean of all check-in locations of that user.



```
def define_alpha():
    alpha = {
        'food': 1.64,
        'nightlife': 1.61,
        'travel': 2.22,
        'work': 1.62,
        'home': 1.62,
        'shops': 1.64,
        'entertainment': 1.64,
        'art': 1.64,
        'parks': 1.68,
        'education': 1.96
    }
    return alpha
```

Figure 2

V. Results

Data Collection

In this study, we have used the Foursquare check-ins dataset from 12 April 2012 to 16 February 2013 in New York city. The dataset is courtesy of Dingqi Yang. The sampled dataset contains 227,428 check-ins and report a quantitative assessment of human mobility patterns by analyzing the spatial and temporal aspects associated with users' footprints.

Data Preprocessing

The dataset was a non-uniform space-separated raw data containing missing values combined with inconsistent indentation. The data preprocessing phase took almost half the time dedicated to the project. We used python to trim extra white spaces and get data to a consistent format. We used regression approach to predict missing values.

System Design

We developed the system using Python, taking advantage of several libraries such as numpy, matplotlib, sklearn, pandas, scipy, surprise, math and gmplot. Our application reads in a CSV file containing preprocessed foursquare check in data, generates visualizations and provides recommendations based on spatial-temporal analysis.

Visualization Results

We visualized our pre-processed NYC Foursquare data frame using gmplot by plotting user longitude and latitude values of user check-ins for different venues to set the initial view of our map. The shaded red region in the heat-map shows venues with peak number of check-ins during a particular timeframe during the day.

Figure 3 illustrates the distribution of various user check-ins on a Google map for different kinds of venues in the city.

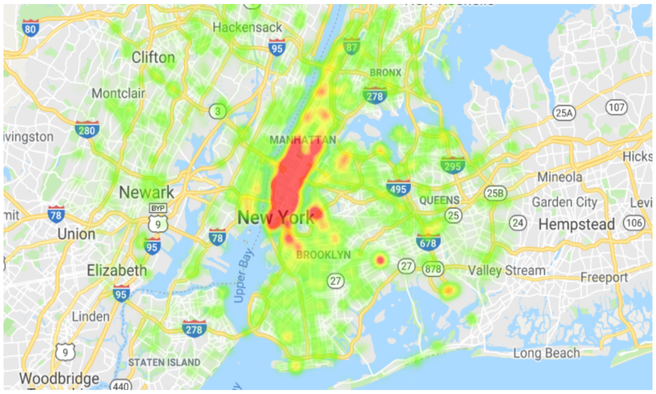


Figure 3

Figure 4 illustrates a histogram for visualizing number of check-ins against check-ins hours during a particular month and day of the year. We observe that there are three clear peaks showing coffee, lunch, and dinner times in almost every timeframe and few peaks ending due to late night bar check-ins. The difference between different timeframes considering days and hours is quite defined.

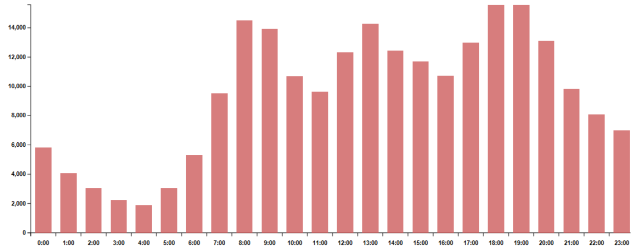


Figure 4

Figure 5 illustrates the distribution of check-ins in the different venue categories, say Bars and Coffee shops, across different days and times of a week. We see that the Coffee shops show a high morning peak, presumably as people get coffee after getting to their workplaces. Comparatively, another venue category, like bars are more useful at visualizing when people leave work and show a peak only after evening hours.

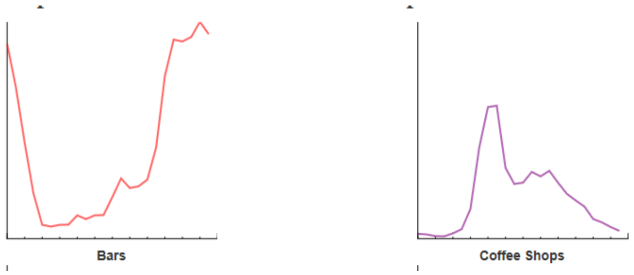
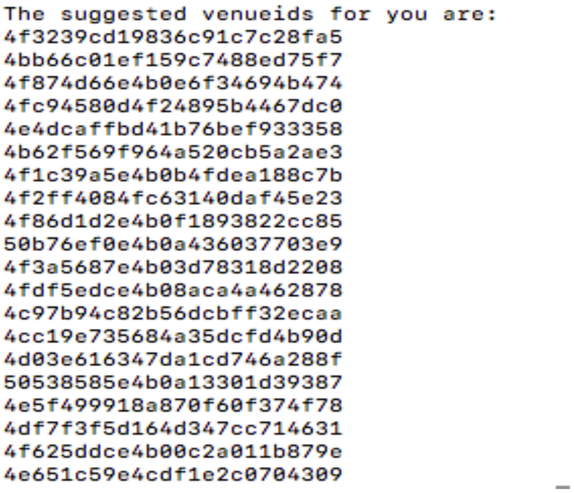


Figure 5

Recommendation Results

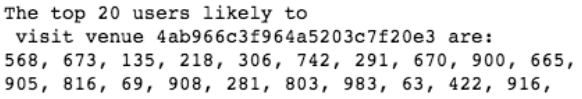
1. **Recommending places to users**
   The probabilistic approach gives the following result for the input: Category: Coffee Shops, User: 642, Time: 16:00 hrs. Since the data contains anonymized venue ids, we suggest venue ids and not the names (Fig 6).



```
The suggested venueids for you are:
4f3239cd19836c91c7c28fa5
4bb66c01ef159c7488ed75f7
4f874d66e4b0e6f34694b474
4fc94580d4f24895b4467dc0
4e4dcaffbd41b76bef933358
4b62f569f964a520cb5a2ae3
4f1c39a5e4b0b4fdea188c7b
4f2ff4084fc63140daf45e23
4f86d1d2e4b0f1893822cc85
50b76ef0e4b0a436037703e9
4f3a5687e4b03d78318d2208
4fdf5edce4b08aca4a462878
4c97b94c82b56dcbff32ecaa
4cc19e735684a35dcfd4b90d
4d03e616347da1cd746a288f
50538585e4b0a13301d39387
4e5f499918a870f60f374f78
4df7f3f5d164d347cc714631
4f625ddce4b00c2a011b879e
4e651c59e4cdf1e2c0704309
```

Figure 6

2. **Recommending users to places**
   The Bayesian Model combined with item-item collaborative filtering approach gives the following results. Again, since the users are anonymized in the data, we suggest user ids (Fig 7).



```
The top 20 users likely to
 visit venue 4ab966c3f964a5203c7f20e3 are:
568, 673, 135, 218, 306, 742, 291, 670, 900, 665,
905, 816, 69, 908, 281, 803, 983, 63, 422, 916,
```

Figure 7

VI. Conclusion

Understanding user spatial temporal activity preference can benefit users by providing them with customized location based services. In this project, we generated a heat-map of all the NYC check ins in the dataset which gives us a view of the most popular places where people visit. Our recommendation system is based on the idea that a user's choice of a venue is not only based on user's desire to go to a popular place but also user's preference on the distance he/she wants to travel. We take these into account in both our models - while recommending venues to users and while recommending users to venues. By mining and visualizing these activity records based on various parameters like check-ins through the day, peaks for venue categories and scatter plot for data points, we mined user's activity preference.

In the future, we plan to broaden this work in several directions. Firstly, our model fails when making recommendations to a new user about whom we have little or no data of their preferences. In such cases we have no way of calculating similarity measures or distance measures and hence our system does not give very accurate results. Therefore, a way to provide recommendations for new users could be devised as an additional task.

Secondly, if we had to determine the approach to this project again, we would spend more time on literature review of similarity measure calculations and mathematics concepts.

Thirdly, we could compare and contrast results of state of the art user-user collaborative filtering algorithms. Also, we would incorporate collaborative filtering to recommend venues to users in model 1 and also find a way to incorporate time factor in model 2.

VII. Repository link

http://tiny.cc/stdm-project

VIII. References

1. X. Liu, Y. Liu, K. Aberer, and C. Miao. Personalized Point-of-Interest Recommendation by Mining User's Preference Transition. In Proc. of CIKM 2013.
2. Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. Magnenat-Thalmann. Time-aware Point-of-interest Recommendation. In Proc. of SIGIR 2013.
3. H. Gao, J. Tnag, X. Hu, and H. Liu. Exploring Temporal Effects for Location Recommendation on Location-Based Social Networks. In Proc. of RecSys 2013.
4. Z. Cheng, J. Caverlee, K. Lee, and D. Sui. Exploring Millions of Footprints in Location Sharing Services. In Proc. of ICWSM 2011.
5. A. Noulas, S. Scellato, C. Mascolo, and M. Pontil. An Empirical Study of Geographic User Activity Patterns in Foursquare. In Proc. of ICWSM 2011.
6. Dingqi Yang, Daqing Zhang, Vincent W. Zheng, Zhiyong Yu. Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs. IEEE Trans. on Systems, Man, and Cybernetics: Systems, (TSMC), 45(1), 129-142, 2015
7. https://en.wikipedia.org/wiki/Haversine_formula
8. http://www.primarydigit.com/blog/-a-location-based-visitor-recommender-system-using-foursquare (Sarvesh Saran, Dept of CS, University of Minnesota)

IX. Contributions

Sanya Kathuria

1. Data Preprocessing
2. Data Exploration
3. Data Visualization Module

Time taken: 60 hrs approx.

Neel Kapadia, Tushita Roychaudhury

1. Data Preprocessing
2. Data Exploration
3. Data Transformation
4. Recommending places to users
5. Recommending users to places

Time Taken: 60 hrs approx.

Every member of team contributed to Data Preprocessing and Data Exploration. Once this was achieved, we split the work into 3 modules. First of which, Data Visualization (Heat map and graphs) were managed by Sanya Kathuria. She worked on google map API for the heat map visualization and used d3 (javascript framework) for data visualization and graphs. She noticed some interesting patterns in the user behavior which were helpful in the implementation of other modules.

The second and third modules, recommending users to places and places to users, were done by Neel Kapadia and Tushita Roychaudhury. They used python to code the modules and various libraries such as sklearn, scipy, numpy, pandas were very helpful. They learnt the mathematics behind Bayesian modelling, item-item collaborative filtering which helped us build a base for application of these algorithms. The idea of using weights w1 and w2 in the module 2 was unique and required many attempts to get to a formula and method to determine the weights such that they are helpful in recommendation. They used convex hull and DBSCAN to trim out outliers and assign weights to the parameter w1. For w2, they applied penalty based on time difference of current time and average time of all check-ins at that venue.