Where will I go next?: Predicting Future Categorical Check-ins in Location Based Social Networks

Velin Kounev, Telecommunication PhD Program, School of Information Science,
University of Pittsburgh, vkounev@pitt.edu

***Abstract*—Models to predict the future location of users have been developed in the past few decades. However, these efforts cannot drive applications related to location-based targeting since they focus on flat geographic prediction with no semantic information. With the emergence of Location Based Social Networks (LBSN) geographical data can be supplemented with contextual information. An efficient location predictor might bring numerous opportunities and commercial benefits. In this work we propose two simple predictors modeling future geo-contextual user behavior. The algorithms have two outputs: first the most likely next visit in terms of category and second the expected time frame, in when, such a visit may occur. The predictors use categorized user activities as unique check-ins at specific times. Using real data obtained from the commercial LBSN (FourSquare), we show the efficiency of the algorithms.**

***Index Terms*—Location Based Social Networks, Future Check-in Model Prediction.**

I. Introduction

Predicting a mobile users location has for years been a research focus in the field of wireless communication. This is due to the fact that accurate tracking can significantly improve performance and user experience in those systems. The goal of most is to provide a better user experience via location aware application design. In the wireless cell phone infrastructure, some predictor algorithms have already been developed. Most of these algorithms are based on segment matching, Order-k Markov, and the LZ-Algorithms clustering. These latter two algorithms are domain-independent predictors, used in a large array of applications. At their core, their task is to find the largest probability for the next user's location, depending upon current location and recent movement history [2] [3]. To achieve the prediction goal, researchers use GPS geo-coordinate data as the main prediction input. Semantic information, such as type of activity of the user, is not available to the predictor.

In contrast, Location Based Social Networks (LBSN), have such information available to them due to the nature of their service. In the past decade LBSN's have become quite popular. FourSquare is one example of an LBSN network that has risen in user popularity. Using this type of a service users can "check-in", thereby indicating their presence in a physical location (such as Restaurants, Work, etc). This information can then be shared with their friends in the network or even published to third party networks (e.g. Facebook, Twitter). Furthermore, these services provide extra incentive for users to check in by offering discounts and deals for the number of times visited or most check-ins in a specific venue. Additionally, FourSquare provides information such as venue type (Restaurant, Work, etc.), location tips, and number of users currently present in a venue [4].

Due to the rapid increase in use of smart mobile terminals, users have embraced the check-in model of such networks. Location prediction is now no longer a simple tracking problem, but an empirical social problem. By solving this problem, researchers can provide additional semantic insight of future user behavior. This insight has tremendous commercial benefit in offering targeted venue incentives to specific users. For instance, a user may have a history of visiting different French restaurants quite often and the check-in history shows that this occurs mostly on weekday evenings. A contextual prediction system may then be used to recommend similar but new restaurant to the user at the proper time, therefore providing service to the user and increasing business volume to the venue(s).

Social science has long sought to classify human behavior according to individual attributes. The main advantage of LBSN is that users are willing to share private activity information readily. From the social point of view, a user might be labeled by his activity preference (food lover, sports lover, etc.). More specifically, the users daily activities during different time periods are tractable based on check-in history. For example, if a user has in the past several months visited a grocery store almost every Saturday morning, he probably will be going to that same store on the upcoming Saturday morning.

COLLABORATECOM 2012, October 14-17, Pittsburgh, United States

DOI 10.4108/icst.collaboratecom.2012.250536

Malmgren et al. [1] show that users tend to send email activities in distinctive patterns. Furthermore, the researchers found that once a user settles into a pattern, changes in behavior are seldom. This fact was explored and the Markov-based prediction model proved to be accurate. Further research [11] in the field indicated that personal diaries were used in order to quantify participant's daily activity patterns. The conclusion presented was that individuals indeed follow set patterns in their daily lives; however, the actual pattern varies from one person to another.

In this paper, we attempt to utilize users pattern activities to predict the next check-in as categorical venue. First, we present a basic straightforward model in order to provide a better understanding as to what a good user behavior prediction is in an LBSN outcome. Second, we present a more robust model in an attempt to improve prediction accuracy. The simulation results for both models are satisfactory, however far from ideal. This is due to the main difference between cellular and LBSN location data. In LBSN, people have the choice to share their location information or not. As such, the location information is not necessarily a true representation of a user's daily activity. It is indeed a "social" representation, as individuals use check-ins in order to present themselves in a socially acceptable way. In contrast, in wireless phone systems the user has no such choice so the location data is more true to the actual motion behavior. As mentioned previously, the problem with cellular location data is that there is no contextual information.

The rest of this paper is organized as follows: section 2 presents a baseline prediction model; section 3 presents a more robust prediction model; section 4 presents simulation results; and section 5 discusses the conclusion and future work.

II. Baseline Model

We start by presenting the baseline prediction model. The predicting algorithm leverages users check-in history in order to predict a future check-in in terms of category and time. The method makes use of statistical analysis to achieve this goal. The model uses a history window that can be adjusted in order to incorporate more or fewer data in the prediction. Besides the users own history, other information can be added to this predictor, for instance, friends information [5], [8], [7], distance a user is willing to travel. This simple algorithm can be a start point of applying social analysis to LBSN location prediction.

The model works in three steps:

1) Period slicing: divide the week into seven days - Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday (total of seven); divide all days into six periods of the day: Early Hours, Morning, Noon, Afternoon, Evening, Night (total of six). This results in forty-two distinct periods of the week.

2) Frequency counting: create a histogram/frequency count for each check-in category in each period of the week. For example: MondayMorningCheckInProbability = 0.023, MondayNoonCheckInProbability = 0.0163.

3) Predicting: based on the counted frequency and time slice information, the algorithm predicts if a given category check-in is likely in the next time slice. For instance, if the next time slice is Tuesday evening and from the user's history the check-in probability for category A is larger than the average weekly check-in probability for the category, then a positive check-in prediction is made. Otherwise, the algorithm assumes that the user will not check-in. For this experiment, the algorithm used a check-in history view window of two weeks - fourteen days. Simulation ran one day at a time. Prediction was made one check-in in advance.

From the simulations we obtained that this model has category prediction accuracy of around 30%. Once time prediction was added, the accuracy of time prediction is close to random. The error varied from 4% to 63% with average of 32%. The performance results of this predictor are not satisfactory.

III. A More Robust Model

In the previous section we discussed the base case linear model algorithm for prediction of future check-ins. Our basic results showed that linear prediction had some success with prediction of future category check-in; however, it struggled to identify a clear relationship in the time series. The next step of the investigation was to find a more robust, non-linear model for prediction.

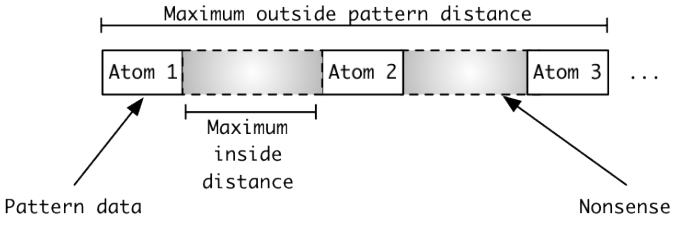

Fig. 1. Aprior Close algorithm parameters.

R. Agrawal and R. Skirant [6] built on this previous data mining work [10] and introduced the Apriori Algo-

rithm. Their algorithm is not focused on finding patterns in the data, but instead on finding sequence of events in a given dataset. Originally, the algorithm was intended to identify the sequence of repeating items bought by customers over a number of separate store visits. In the algorithm items are identified by an ID number and transaction is composed of one of more item(s). In order for two items to be considered as part of a sequence, they do not have to be one right after another. In other words, sequence item one can be followed by sequence item two immediately, or there can be a gap between them containing other non-sequence ”nonsense” items. One drawback of this algorithm is that it does not have a notion of time elapsed between item one, two, three and so forth. The only certainty is that items in the sequence occurred in sequential order. Patterns discovered can be of any length, with a minimum length of two.

There are three parameters to the algorithm: minimum pattern support, minimum inner pattern distance and maximum other pattern distance. The minimum patterns support specified as percentage, indicates the minimum percentage of transaction that need to contain a pattern, before the pattern is considered of a valid support. Minimum inner distance indicates the maximum distance between two consecutive patterns elements, and maximum outer distance indicates the maximum distance between the beginning and ending elements of a pattern. Figure 1 present the basic algorithm model and its parameters.

We used the exact same approach to extract sequence of check-in from user activity histories. By identifying patterns, our system could detect the beginning of such known check-in sequence and predict what was most likely to occur. The detailed steps of the algorithm we used are presented below.

A. Pattern Indentification Phase

The first step we took was to encode all user check-ins in order to use the Apriori algorithm to discover patterns in the dataset. The entire history of check-ins for a single user was considered to be one transaction. In other words, if fifty user histories are used, then there will be fifty transactions. Each user check-in is encoded by using an integer from 1 to 9, representing FourSquare’s top level categories: Arts Entertainment, College University, Food, Professional Other Places, Nightlife Spot, Great Outdoors, Shop Service, Travel Transport, Residence. Ten percent of the user data was used as a training set in order to identify common patterns between users in the population. Using SPMF data mining framework [12], common patterns were discovered between users. The question we attempted to answer was: If a user check-in is in Category A, does this tell us anything about any possible future check-ins? As such, the algorithm tries to discover similar check-in patterns between all users.

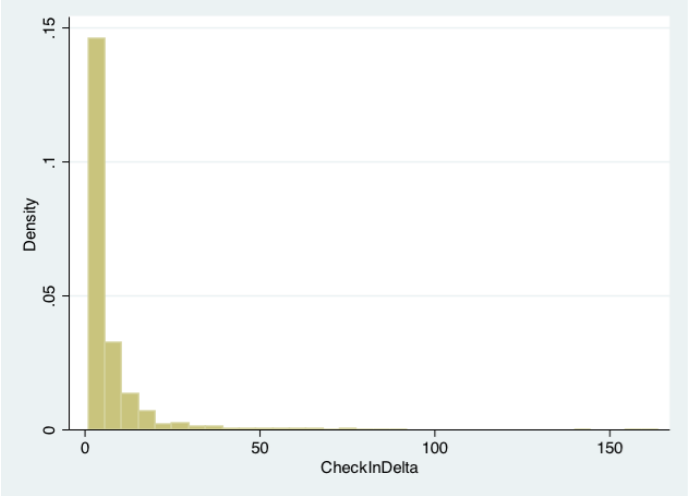

Fig. 2. Histogram time difference between elements of pattern ’Work followed by Restaurant’.

B. Algorithm Training Phase

For our second step, we attempted to put a time frame between different pattern elements. For instance, many users have a check-in in Category A and a few check-ins later into Category B. We wanted to identify what the expected check-in delta between check-ins one and two was. The histograms of inner check-in delta between a few categories is presented in Figure 2. As we can see from the histogram of time delta between check-ins, the delta is a long tail distribution. This holds true for all other delta time points. One of the most interesting observations is that the time statistic from FourSquare data seem to follow such a distribution. This fact makes time-based prediction very difficult.

C. Prediction Phase

The predictor model is presented in Figure 3. After number of possible user population patterns are identified via the training phase, those patterns are used on individual histories. Each user check-in can trigger one or more patterns that indicates future check-in. After a pattern is indicated to be active, a possible future prediction with valid timeframe is set as predicted. If the predicted check-in occurs in the valid time frame, then the prediction is considered accurate. It is important to

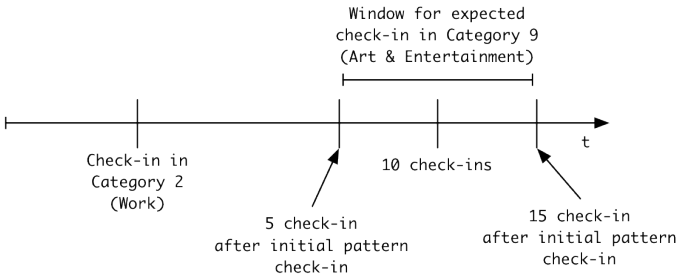

Fig. 3. Apriori Close based check-in Predictor.

point out that, the check-in time frame has minimum and maximum valid time window, as show in the Figure 3.

IV. Simulation

A. Dataset

We used the data obtained from Cheng et. al. [9]. The authors used twitter API to crawl FourSquare check-ins published at large. Twitter messages support the inclusion of geo-tags (latitude/longitude) as well as FourSquare location sharing services. Each check-in in the dataset had a geo-tagged status update and timestamp. We further obtained corresponding category towards each update through FourSquares open API, in order to categorize each check-in. FourSquare supports several categorical levels, however we only used the primary top categories as reference.

The location crawler in [9] ran for almost 4 months, resulting in a total collection of 225,098 users and 22,506,721 unique check-ins. In running a prototype of the predictor we crawled close to 6,500,000 check-in records that had valid URLs of FourSquare categories. We created a database to store the relations of all the crawled information. The useful check-ins are tuples: userID , tweetID, text, location, time, categoryID, category name. The predictor was applied on all 6,5 million check-in records and evaluated the accuracy it achieved. Each check-in in the database belonged to a user that has at least 10 check-ins.

B. Setup

The experimental data subset contained 6.5 million check-ins. Furthermore, an open-source framework SPMF [12] was used to discover patterns using Apriori Close algorithm. For the purposes of training we used 10 percent of the data, or roughly 650,000 check-ins. The Apriori Close Algorithm ran with a threshold set at 25 percent (the minimum number of users that have to display a check-in pattern, for a pattern to be considered valid). For instance, pattern 2 4 (Work, Restaurant) had the support of 54 percent of all users, and pattern 2 9 (Work, Art Entertainment) was supported by 30 percent of the users.

The next step of the training process was to find what is the mean check-in delta between atoms in the discovered patterns. The same training set was used to calculate the statistic. For instance, in using an exhaustive search of the training set it was found that there were on average 6 check-in delta between atom 2 and 4 in pattern 2 4. However, the variance of the delta is also quite large 15 check-ins.

The rest, 90 percent of the data, was used to measure the performance of the prediction algorithm. The algorithm started by using the pre-calculated statistics for prediction. Once the algorithm encountered the first atom of a pattern, it sets the pattern as active. Next, was to predict when the second atom would be encountered. For instance, once the algorithm saw check-in in category 2 (Work), it expected to see a check-in in category 9 (Restaurant) within a mean of 6 check-ins, or a check-in into category 9 (Art Entertainment) within a mean of 10 check-ins (assuming that both patterns 2 4 and 2 9 were active).

Each expected delta check-in mean had a upper and lower window threshold, calculated as the variance of the mean. For instance, the second check-in in category 9, was expected within 10 check-ins, plus or minus 5 check-ins. The timeframe window for the pattern's second atom was set between check-in 5 and check-in 15. If the second check-in occurred in this window, then the prediction was considered correct and within a valid check-in timeframe. If however, the second check-in occurred outside the window, either before the lower window bound or after the upper window bound, the check-in prediction was considered correct, but outside the valid time.

Obviously, a large prediction window is of no use due to its poor accuracy. Therefore, the algorithm corrected the delta mean and the upper and lower bound after each prediction, attempting to shrink the window size . The measurement accuracy of the algorithm was then not only the correct vs. incorrect check-in prediction, but the average size of the prediction window. A smaller window size and correct check-in prediction accuracy were the main performance measurements of the prediction algorithm.

Statistic	Test Run 1	Test Run 2	Test Run 3
Apriori support threshold	10%	20%	30%
Total checkins	221588	221588	221588
Total patterns	352680	306828	246352
Average patterns per user	61	53	42
Total predicted patterns	239106	214795	179335
Total predicted patterns within timeframe	91051	81928	67880
Percentage of check-ins predicted	107%	96%	80%
Percentage of check-ins predicted within timeframe	41%	36%	30%
Average check-in timeframe window size	5.3	5.2	4.8
False positives	38%	22%	15%

TABLE I
CHECK-IN PREDICTION SIMULATION RESULTS. PLEASE NOTE: PERCENTAGE OF CHECK IN PREDICTED CAN EXCEED 100%, DUE TO THE FACT THAT TWO OR MORE PATTERNS CAN PREDICT THE SAME CHECK IN.

C. Results

The main parameter to the simulation is the Apriori Algorithm minimum pattern support threshold. By varying this parameter the predictor can predict greater or fewer check-ins. However, greater predictions come at the expense of false positives. Results from three separate experiments are presented in Table I. The three experiments ran with algorithm thresholds of 10%, 20% and 30%. For the purpose of comparison, each test run presented was stopped exactly after the same amount of check-ins, indicated by the second row in the column.

Due to different Apriori threshold parameter, the pattern discovered for each test run was different. As expected the least stringent run, test run 1, had the most possible total pattern, the most average patterns per user - 62, and also predicted the most check-ins. It is important to point out that the total number of predicted check-ins in row 5 is greater than the total number of check-ins of the users. This is due to the fact that one check-in could be predicted by multiple patterns. For instance, the second atom in patterns 2 9 and 4 9 in both cases is category 9. If indeed the user checks-in in category 2 and sometime later in category 4 (before a check-in in category 9), then both of those patterns become active. The algorithm predicts that a check-in in category 9 is likely, and if it occurs, then two patterns were closed successfully.

The most important performance statistics are presented in the last three rows, the total percentage of correctly predicted check-ins within a time frame, the average size of the timeframe, and the number of false positives. Ideally, we would like to see high positive prediction and small timeframe window, together with low percentage of false positives. False positives are defined as the number of patterns identified and set active by the algorithm, and consequently, not closed by subsequent check-ins.

In the first test run, the correct prediction rate is 41% with timeframe window of 5.3 check-ins. However, the number of false positives is comparable to the correct prediction rate - 38%. Test runs 2 and 3 displayed most lower false positives, however due to the lower number of patterns, the correctly predicted check-in also is lower. It is interesting to point out that the average timeframe window does not seem to vary much in the three cases. We believe this is due to the distribution of any time statistics. It appears that users do not follow any time cyclic pattern in regards to their check-ins. This is indeed the most significant discovery of our research.

Our finding shows that check-in in FourSquare contradict the expected norm in Social Sciences of cyclic human behavior patterns [13]. We suspect that this is due to the fact that users' check-ins does not represent their true mobility pattern, but rather it is a social representation.

V. CONCLUSION

In this paper we presented two simple prediction models for FourSquare future check-ins. The accuracy of these models is good, but not perfect. We believe this is due to the nature of Location Based Social Networks. Future interesting research should include not only data from LBSN, but also possibly data from cell phone carriers. Cellular data will enhance the predictor since

it will have a more complete view of people mobility. Furthermore, we would like to exploit friendships as indicators of possible future check-in locations since user has been shown to be spatially consistent [14].

REFERENCES

[1] R. Dean Malmgren, J. M. Hofman, L. A.N. Amaral and D. J.Watts, Characterizing Individual Communication Patterns *KDD09*, Paris, France.

[2] C. Cheng, R. Jain, Location Prediction Algorithms for Mobile Wireless System September 10, 2002.

[3] L. Song, D. Kotz, R. Jain, X. He, Evaluating location predictors with extensive Wi-Fi mobility data 2004.

[4] A. Noulas, S. Scellato, C. Mascolo, M. Pontil, Exploiting Semantic Annotation for Clustering Geographic Areas and Users in Location-Based Social Networks Association for the Ad- vancement of Artificial Intelligence, 2011.

[5] L. Backstrom, E. Sun, C. Marlow, Find Me If You Can: Improving Geographical Prediction with Social and Spatial Proximity North Carolina, USA, 2011.

[6] R. Agrawal, R. Srikant, Mining Sequential Patterns, IBM Almaden Research Center.

[7] S. Scellato, A. Noulas, C. Mascolo, Exploiting Place Features in Link Prediction on Location-based Social Network *KDD11*, August 21-24, 2011, San Diego, California, USA.

[8] M. Zhou, P. Krishnamurthy, Y. Xu, L. Ma, Physical Distance Vs. Signal Distance: An Analysis Towards Better Location Fingerprinting, 2012.

[9] Z. Cheng, J. Caverlee, K. Lee, D. Z. Sui, Exploring Millions of Footprints in Location Sharing Services Association for the Advancement of Artificial Intelligence.

[10] T. G. Dietterich, R. S. Michalski, Discovering patterns in sequence of events. *Artificial Intelligence*, 25:187-232, 1985.

[11] K. Vrotsou, K. Ellegard, M. Cooper, Everyday life Discoverries: Mining and Visualizing Activity Patterns in Social Science Diary Data, *International Conference Information Visualization*, 2007

[12] P. Fournier-Viger, A Sequential Pattern Mining Framework. *http://www.philippe-fournier-viger.com/spmf/*

[13] J. Cohen, P. Cohen, Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences, Second Edition. 1983.

[14] K. Pelechrinis and P. Krishnamurthy, Location Affiliation Networks: Bonding Social and Spatial Information, *ECML/PKDD 2012*, Bristol, UK, 2012