Location-based Predictions for Personalized Contextual Services using Social Network Data

Rui Zhang, Bob Price, Maurice Chu and Alan Walendowski

Palo Alto Research Center Inc. (PARC, a Xerox Company)

3333 Coyote Hill Rd. Palo Alto CA 94304
Rui.zhang@parc.com

Abstract. User activity prediction can enable powerful personalized services such as target advertising, contextual recommendations, and advanced reminders and social coordination. Predicting user activities is non-trivial for many obvious reasons. This paper focuses on predicting future user visits to various venues based on past visit history. Our approach is most notable in its efficiency and the actionable predictions it produces for recommendations. Experiments based on a large Foursquare data set shows that our approach has an accuracy of up to 76% compared to the 40% from naïve methods, and also produces up to 10x less false alarms. Our experiments unearthed some other interesting insights and inspired several promising research problems.

1 Introduction

Activity prediction is a new technology that has applications to targeted messaging, recommender systems, transportation optimization, home heating energy savings, and social coordination. The idea is to analyze data from several sensor streams (GPS, Email/SMS, audio, web pages viewed, etc.) and use them to update a user model that predicts what the user will do next from patterns observed in past data. In the context of targeted messaging and recommendation, the prediction can be used to pro-actively push information related to the predicted user activity at the right moment ahead of time. For instance, we may predict that a business traveler is likely to travel to a destination in 3 weeks and that he is likely to purchase air tickets 2 weeks ahead of his travel (i.e. within 1 week). In this case, we may want to push an alert (e.g. via email, text message or phone pop-up) to the user in 1 week with an airline discount offer that is attractive to the user. This proactive push is especially useful given that most companies book discount travels for their employees, and once the traveler commits to a ticket, it is non-refundable (even if there is a much cheaper discount available from another airline).

Herein location check-ins are used in place of activities for inference. There is a close relationship between place and activity [2], and many techniques that are applied to only predicting place can be easily extended to predicting activity. This paper thus specifically focuses on location prediction, in order to facilitate making the aforementioned hyper-personalized recommendations. This entails solving two challenging problems: (1) predicting the probability of future activities, and (2) triggering a recommendation action based on the computed probabilities (most prominently if it crosses a certain threshold).

The former is challenging in two respects. First, there might not be much history available per user per location to find meaningful patterns. In a realistic application, the prediction has to be made in real-time for thousands or millions of users with many activities per user, rendering the cost of computation a potential concern.

The latter is required because recommender systems must act on certain predictions about whether a future activity will take place. Choosing the right threshold of activity prediction certainty can have significant impact on the effectiveness of a recommendation. If the threshold is set too low, many low probability activities would result in irrelevant recommendations being made; conversely, if it's set too high, many relevant recommendations may not be made. Even in cases where the recommendation system is trying to choose the most probable alternative from a number of alternatives, one can argue that the recommendation system should choose to take no action, if the probability of all alternatives is below the optimal threshold.

While there is much relevant work in this popular space, little has been done to address the challenges set forward above. This paper is our attempt in bridging this gap, initially focusing on predicting future visits to venues using the four square data set as a reference.

In the process, we have made the following research contributions:

- An efficient method for predicting the probability of location visits.
- A utility-driven method for triggering contextual recommendations based on the predicted probability
- Empirical validation and findings using a real-world, large-scale data set: the public four square check-in data [7].
- Identifying a number of related research problems that merit further investigation.

The remainder of the paper is organized as follows: the sequel overviews the four square data set referenced throughout this paper. Section 3 discusses our method of location prediction in detail. Experiment setup and results are described in Sections 4. Section 5 reviews related work, before we conclude and outlines a number of promising pieces of ongoing and future work in Section 6.

2 Foursquare data

Our data consists of more than 180 million public check-ins by Foursquare users from April 2010 through mid-2011. We are continuing to collecting more data and have plans towards making this data available in the future.

As illustrated in Figure 1, as users check-in in Foursquare, we listen to corresponding public broadcasts in Twitter [8] and build a database of check-in events over time.

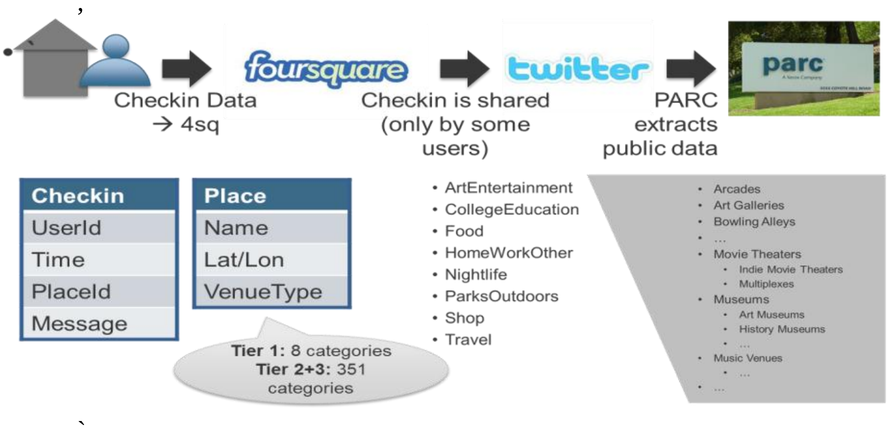


Fig. 1. Foursquare data overview: collection and schema

2.1 Data collection and filtering

When a user 'checks in' to a location with Foursquare, they are given the option to publicly broadcast the check-in via twitter. We had a Twitter filtered search set up to listen for tweets that contained text strings that might indicate a Foursquare check in. The default check-in message usually contains a URL (e.g. "I'm at HSBC (Plaza Fiesta, Merida) http://t.co/svYDhTW2"), so we listened for "4sq.com" and "t.co".

To capture the public check-in tweets, we ran a single Java process that connected to Twitter's Streaming API and registered interest in a set of search terms as described earlier. For expediency, and because of the potential burst nature of any Twitter feed, the raw JSON text of the tweets was simply saved to log files. Periodically, usually once per day, the contents of any new log files were decomposed into relevant fields (Twitter userID, tweet ID, embedded URL of check-in, etc) and imported into a MySQL database. Some tweets did not resolve to a check-in or lead to a venue. Those tweets were subsequently ignored in the database.

The vast majority of users that publicly tweeted their check-ins only did so a few times so, for our purposes, we selected a subset of users that had each checked in at least 500 times. This gave us roughly 8800 sets of check-ins to work with, comprising ~6.7M check-ins to ~967K venues.

2.2 Data description

Each check-in event consists of a unique ID, an ID of the user who checked in and further information about the venue, including the full name, street address, city, state, postal code, country, latitude, longitudes, and a tier 1 Foursquare-assigned category and a tier 2 category (e.g. "Food/Coffee Shop"). There are a total of 8 Tier 1 categories as enumerated in Figure 1 and numerous tier 2 categories per tier 1 category.

3 Methodology

Accurately predicting the fine grained behavior of individual users can be very difficult. The prior evidence available for any one user about events for a specific day, time and tier 1 and 2 venue type will often not be sufficient to make accurate predictions. This is particularly true for new users that have little history with the system (i.e., the cold start problem). This sparseness is apparent in the activity history plots taken from our Four Square dataset. A plot for a single user is shown in Figure 2.

We address the data scarcity problem in part by ignoring the specific time of day and predicting only the venue type (e.g., generic restaurant). Unfortunately, the data will still be too sparse in many cases for new users. A standard statistical technique in such situations is to make use of priors. In this case, we opt for the conceptual simplicity of an Empirical Bayes formulation [10]. As we will explain, this choice also supports our desired architectural decomposition. We first model the venue visit probabilities for the population as a whole. So on average, we might discover people are most likely to visit restaurants around Noon and 6:00 p.m. We then use the maximum likelihood estimates for the population visit probabilities as priors for individual level models. In the absence of additional information, the individual would be predicted to follow the population trends. However, if there is specific evidence for the individual, this evidence could override the population. So, if an individual likes to go out with friends on Fridays to a very fany restaurant, the record of their actual behavior would cause their predictions to diverge from the population on Fridays.

As explained in Section 1, modeling the visit probability is only the first step. Given the probability of a visit, we must decide whether the strength of the prediction warrants taking an action (such as making a recommendation, sending a coupon, or initiating a survey). The details of our prediction model and the procedure for tuning the decision threshold are described below.

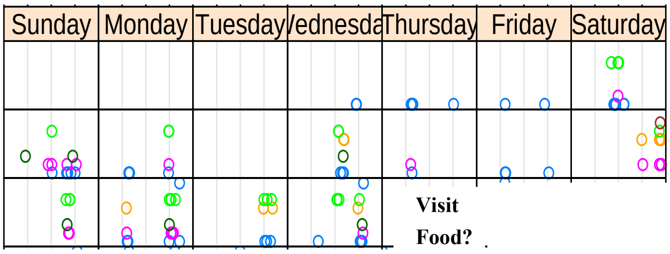


Fig. 2. Example of Four Square visitation data for an individual user and illustration of the problem of predicting visit on a given day. Different colors represent different venue types.

3.1 Logistic regression of contextual factors

We choose to use logistic regression models for both individual and population level predictions. Logistic regression is attractive for our problem, because it's computational inexpensive to evaluate probabilities at run time and the shape of the model appears to fit the patterns we can visually see from the data well. We use logistic functions of the form

$$\text{logit}(p) = a + b_1 * x_1 + \ldots\ldots + b_n * x_n$$

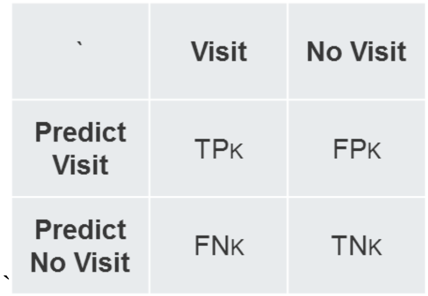
where a is the intercept x_i represents the value of the i^{th} feature, b_i represents the learned weight of the i^{th} feature and p is the probability of a future activity taking place given the feature variables.

Categorical variables, where there is no natural ordering, are represented by a set of binary variables with one variable for each category. Each binary variable takes the value 1 when the categorical variable is in its corresponding state. For instance, the day of the week feature is converted into 7 binary feature variables {isMonday, isTuesday. IsWednesday, isThursday, isFriday, isSaturday, isSunday}. In our model, this categorical feature is augmented by a numerical variable daysSinceLastVisit. The intercept and the weight parameters for each feature must be learned. For both the intercept and the parameter weights we specified essentially an uninformative prior (a normal prior with mean 0 and variance $1x10^6$). We then used Gibbs sampling [10] to estimate the parameters for the population from the data. This then gives us a model for predicting the visitation behavior of an average individual drawn from the population as a whole.

A key tenant of personalized recommendation, however, is that each individual is different. To accommodate individual differences, we learn individual models. Following the empirical Bayes paradigm, we use the maximum likelihood values from the population model parameters as the prior mean for individual level models and a variance of 0.001 for the individual level model prior variance. This causes individuals to generally fall in the range of the population probabilities unless the individual provides significant evidence of deviations from population behavior.

3.2 Setting the triggering threshold for predictions

Given a probability of a visitation, we must decide whether the recommender should take action. A common method of making this decision is to set a threshold K and decide to take action whenever the probability of a visit exceeds K. The optimum value for K can be found by defining a utility function and searching for the threshold K that maximizes expected utility. As shown in the confusion matrix below in Figure 3, a prediction can result in one of four possible outcomes. Given a specific K, each of these outcomes will have a probability. Thus K determines the trade-offs between these outcomes.



`	Visit	No Visit
Predict Visit	TPK	FPK
Predict No Visit	FNK	TNK

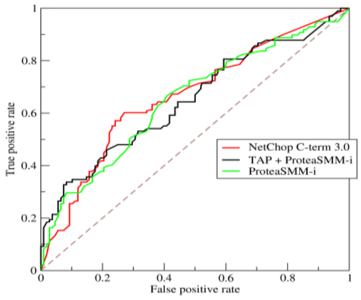


Fig. 3. Given a probability threshold K, the confusion matrix describes the probability of a true positive TP_K, false positive FP_K, false negative FN_K, and true negative TN_K (left). The trade-offs amongst any two of these four probabilities are often captured by an ROC curve (right).

In this work, the expected utility function has the form:

$$U(K) = a_1\,(TP_K) + a_2(FP_K) + a_3(FN_K) + a_4(TN_K)\ .$$

The a_1 term represents the value to the user when we correctly predict their visiting behavior and would be generally be positive. The term a_2 represents the case where we believed they would visit but did not. In the case of pushing coupons this could result in negative value as we have bothered the user with a coupon that was irrelevant. Similarly, the a_3 and a_4 terms represents the cost associated with these outcomes. The optimal threshold is simply:

$$K^* = \mathrm{argmax}\ U(K)\ .$$

First, a K must at a minimum ensure that the decisions we make are superior to a designated baseline method where no prediction is used. For example, one baseline may be always predicting there is a visit, or nor a visit.

Second, we seek a K$_{.}$ that yields a high (if not maximum) utility score. While a brute force approach is in principle possible, it can be expensive given a large number of possible values of K$_{,}$, especially considering that the search process has to be repeated for the number of users and the number of venues per user).

We adopt a greedy approach, where we first search the entire space [0,1] at a coarse granularity to determine a promising range around the K with the highest utility score which also satisfies the baseline criterion above. We then iteratively search the promising area at a finer granularity; and so on until the granularity is deemed sufficiently fine. If at acertain iteration, the highest utility score is no better than baselines, we redo the search of the entire space using a finer granularity (vs. focusing on a promising local area) in the hope of finding a threshold value that outperforms the baseline. An example heuristic is outlined below.

```
DEPTH = 0;
$K_{start}$ = 0; $K_{end}$ = 1
1. P = {$K_{start}$ , $K_{start}$ +delta, … , $K_{end}$ -delta, $K_{end}$} , DEPTH = DEPTH + 1;
2. FOR all K in P,
Compute the corresponding TP, FP, FN, TN and  U(K );
3. Find the K* in P with the highest utility score,
4. IF U (K*) >= U(Baseline) THEN GOTO Step #5
             ELSE _delta  = delta / 10; GOTO Step #1.
5. IF DEPTH > 2 THEN EXIT,
                ELSE GOTO Step #6
6. delta = delta / 10;    // Increase search granularity,
7. $K_{start}$ = K*- 5 * delta; $K_{end}$ = K*- 5 * delta;
8. GOTO Step #1
```

4 Experiments

We evaluated our approach against the four square data set. The goal was to show the efficacy of our approach and to learn lessons on what could help us do better.

4.1 Software and Hardware setup

We implemented an end-to-end prototype for the approach detailed in Section 3, including (a) automated data pre-processing, (b) automated model learning, (c) automated prediction and result evaluation/visualization. The implementation was done in a combination of R, Jags [10] and scripts.

We performed experiments using the prototype on a compute server running Ubuntu 10.04 LTS. It has 8 Intel Xeon E5640 CPUs, each with 4 cores running at 2.67GHz, 16Gb of physical RAM and 300Gb of swap space.

4.2 Evaluation Method and Metrics

We divide our data into a training set and a test set. For our population-level predictions, we use the model to predict visits of users that are not used in training. At the individual level, since there is often sparse data regarding the visits of a particular user to a particular fine-grained venue, we use a leave-k%-out strategy to estimate the TP, FP, FN, TN given a set of training data. The first (100 – k) % of the data will be used for training and the last k% used for testing and computing the TP, FP, FN, TN.

In order to reveal more interesting insights, we contrasted three different visit probability models: (1) weekday, (2) daysSince and (3) weekday + daysSince. They respectively use weekday, daysSince and the combination as contextual features in the logistic regression. Two baseline methods, namely AlwaysVisit, NoVisit, were used, where the baseline always predicts there is or is not a visit, respectively.

We use the following metrics to gauge the quality of prediction:

- Accuracy = TP + TN / total
- Precision = TP / (Predict Visit)
- True Positive Rate (aka Recall) = TP / Visit
- False Positive Rate = FP / No Visit
- F-Score = 2*Precision*Recall / (Precision+Recall)

More business driven metrics for contextual service applications can include advertisement click-through rate lift and Return-of-Investment (ROI) lift. We are not in a position to measure these with the four square data set, but plan to do so by integrating our approach with a live application in the future.

4.3 Overall performance results

We compared the overall performance of our models against the baseline models, on average, across a subset of 2000 users and across all venue types defined by Foursquare. Figure 4 presents the results of this comparison. The most notable and business-relevant highlights are:

- At the population level, our approach is up to around 2x more accurate vs. baseline (76% vs. 40%) and produces around 10x less false alarms (8% vs. 100%)
- At the individual level, our approach is up to around 1.5x more accurate vs. baseline (58% vs. 40%) and produces around 2.5x less false alarms (43% vs. 100%)

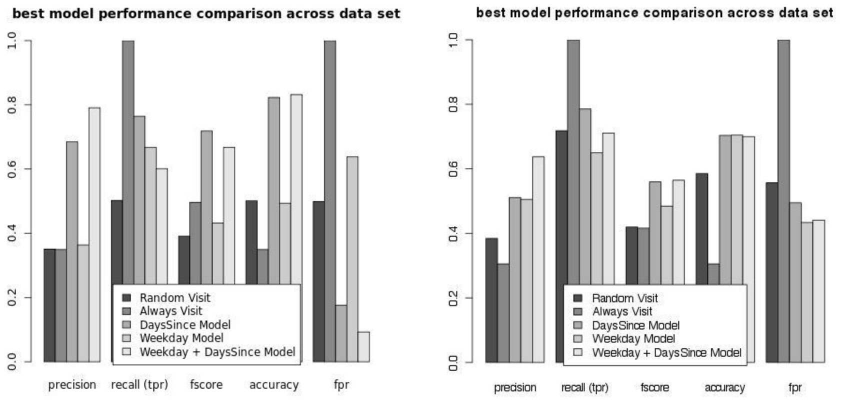


Fig. 4. Overall performance – population (left), individual right). Each bar group compares 3 of our models against 2 baselines regarding a different metric. A higher score is better for the first 4 metrics, while a lower score is better for false positive rate (fpr).

It is interesting that the weekday model performs considerably worse than the days model overall (considerably worse at the population level). This prompted us to investigate the difference in the next subsection. Furthermore, using more features does not guarantee better results. (e.g. see weekday and days vs. daysSince alone in terms of recall and precision in the population case).

4.4 Model sensitivity to venue types

In order to unearth the reason behind the dramatic difference between weekday and daysSince model performance seen in Figure 4, we compared weekday and daysSince model performance for each venue type and observed that the daysSince model was far superior in most venue types, but for some venue types, the weekday model was more accurate. Figure 5 shows two venue types where the latter is the case. This can be explained by the that the weekday is a powerful predictor when it comes to places like hotels and bus stops where traveler activities tend to exhibit strong weekday patterns like the hotel being more busy during the week than on weekends.

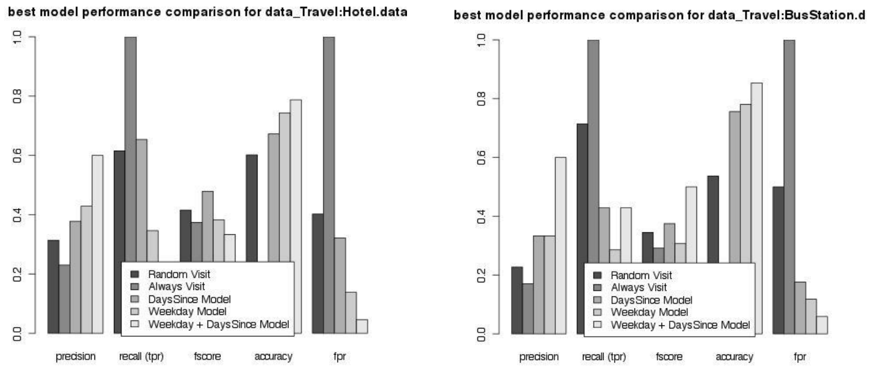


Fig. 5. Results for two example venue types where the weekday model is more accurate than the daysSince model. The style of presentation was the same as in Figure 4.

4.5 Model sensitivity to training set size

We also explored the sensitivity of our approach to training set size. This is particular relevant at the individual level, as not much data may be available for certain individuals and/or venues. Figure 6 illustrates the overall sensitivity of the daysSince and weekday models, average across all venue types and different users. The weekday model is more sensitive, because there are more logistic regression parameters to learn due to the categorical variable conversion in Section 3.2. The sensitivity is higher when there is little data available, but less so where there is more data.

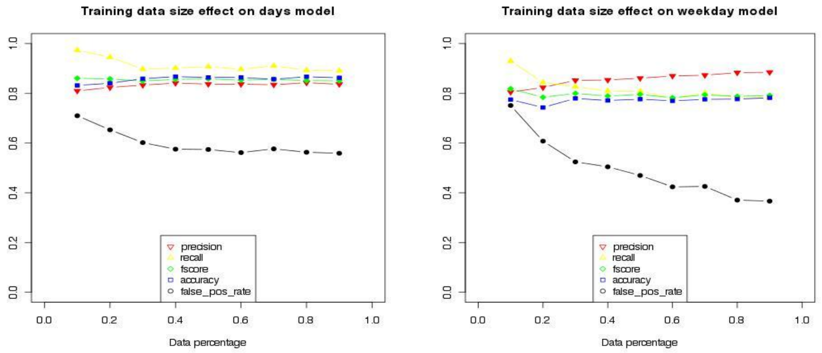


Fig. 6. Overall daysSince and weekday model sensitivity to training data size

5 Related Work

While many projects have looked at location as a context variable to be conditioned on, relatively few projects in the pervasive community have looked at estimating the probability of location visits themselves. Estimating location visit probabilities however, is key to accurately targeting services appropriate to future user locations.

Partridge [2] points out that the problem of training a classifier to predict user activities from GPS traces inherently faces the challenging problem of sparseness and ambiguous labeling. Partridge exploits detailed activity census data collected for government social services work to get around the sparseness and lack of labels in GPS data sets. In our case, however, we need to predict actual locations in the real world. Activity labels, or even abstract locations such as 'workplace' are insufficient.

Like our work, the CLAR system [6] tries to predict user behavior from GPS traces but the behavior of interest for them is the user's activity given a location or a location given an activity. They do not deal directly with the problem of trying to infer the probability of a visit to a location. The GM-FCF system [5] directly makes location aware recommendations to users using a novel combination of social relations and geographic information. Again, although they exploit location to improve recommendation, the probability of visiting a location is not explicitly inferred. The Magitti recommender system [1] incorporated mechanism to infer user activities from unlabeled GPS traces. Using many observations of the user over time, a non-parametric nearest neighbors approach could extract the significant attributes of regions over time in order to extract a model of what type of activity the user prefers at a given time (say eating on Fridays at 1 pm) and well as important attributes of this activity (prefer Asian food, restaurants with patios and non-smoking venues). Again, the emphasis here was on predicting activity preferences rather than the probability of visiting a particular location.

The FLAP system [4] exploits Twitter users whose location is known with high accuracy to predict the location of friends whose location is unknown. Message overlap, collocation features and the overlap of friends lists are used in a Markov Random field to infer the friendship graph. Given the friendship graph a dynamic Bayes net in which the users past location, the location of all the user's friends, the time of day and the status of the day (work/free) is used to predict the user's current location. The authors demonstrate the method on twitter populations from LA and New York and show that the model predicts user locations quite accurately even for users with no explicit location data (54%). While these results are impressive, exploiting social relationships is not practical in our intended application area. In the NextPlace system [3], repeating temporal patterns associated with individual locations are extracted by non-parametric time series analysis. To predict a visit to a location at a time t, a vector of length *m* preceding the time to predict is extracted. This prefix is then matched against all possible previous positions in the history for this location. The authors use the best match to predict the time of the next visit. The NextPlace model can capture the fine micro structure of sequences of events predicting a visit to places, but the prediction model does not take full advantage of probabilistic representations to combine data. It might be possible to combine our hierarchical statistical model with such as system to increase accuracy.

6 Conclusions and Future work

This paper has presented a method for predicting future user visits to venues, for the purpose of providing better contextual recommendation service to the user. Rather than a complex model, we opted for a simpler, efficient logistic regression model that proves reasonably accurate against a real-world data set. Instead of focusing our efforts solely on the probability model (as is in many related works), we proposed a novel way to configure the model with the aim of providing the optimal decision input into a contextual recommendation environment.

We have conducted experiments against the four square data set, including empirical validation of the accuracy of our approach, and empirical understanding of our approach's sensitivity to venue type and training data set size.

Our experience with the Foursquare data has enlightened us to a number of ideas:

- **Application integration:** We aim to integrate the prediction with a prototype coupon recommendation engine, so that coupons may be recommended to users at the right moment ahead of time. This capability may dramatically improve user acceptance of the offer, as people can now receive the recommendation when there is still time to change their plans and actions (for example, switching to an airline with a discount offer before they have already committed to a non-refundable ticket from some other airline. The integration would also allow us to measure more business-driven metrics such as advertisement click-through rate lift and Return-of-Investment (ROI) lift.
- **Segmented model:** The current embodiment is a method of determining a generalized decision tree (also known as logistic regression tree), in that the leaves of the tree are logistic regression models (vs. a prediction value). The input to the model is a set of feature variables, including contextual factors (such as time of day, location etc) and past activity history (such as time since last visit to a venue) from the same or other users, the output of the model is the probability of an activity for a given user.
- **Scalability:** The kind of analytics described in this paper often need be conducted in a real-time manner. This problem is particularly challenging as it scales, that is, when we are continuously collecting and analyzing data about millions of mobile users from dozens or even hundreds of sensors and providing service. The newer generations of phones are becoming increasingly powerful processing platforms. In the mean time, the emergence of cloud infrastructure provides economical and vast compute and storage resources for conducting advanced analytics at large scale. We feel that uniting these two trends can address the challenge of large-scale user activity analytics, via a hybrid architecture that spans both cloud and mobile devices.
- **Privacy:** Population level predictions can only happen naturally in the cloud where data from all users are gathered. Data privacy is a critical issue, as cloud infrastruc-

ture is often controlled by third-party providers and security and privacy mechanisms for the cloud are far from mature. We plan to explore sampling techniques to minimize the number of users having to submit data to the cloud.

- **Prediction at finer time granularity:** Time of day can be a useful indicator of whether the user will check into a particular place. Visual examination of the foursquare data shows that some places (e.g. restaurants, shops) are usually visited during daytime or evening hours. Other places (like home) can be visited at other times. Building a model that can effectively predict the times at which a user checks in is challenging. It seems non-parametric methods such as bucketing may be easiest. Perhaps more desirable would be to use a model with a small number of parameters.

7 References

1. Bellotti, V., Begole, B., Chi, E., Ducheneaut, N., Fang, Ji., Isaacs, E., King, T., Newman, M.W., Partridge, K., Price, B., Rasmussen, P., Roberts,M., Schiano, D. J., Walendowski, A. : Activity-based serendipitous recommendations with the Magitti mobile leisure guide. In: CHI '08 Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems (2008)
2. Partridge, K., Golle, P. : On Using Existing Time-Use Study Data for Ubiquitous Computing Applications. In: UbiComp '08 Proceedings of the 10th international conference on Ubiquitous computing, 2008
3. Scellato, S., Musolesi, M., Mascolo, C., Latora, V., Campbell, A.T. : NextPlace: A Spatio-Temporal Prediction Framework for Pervasive Systems. In: Pervasive (2011)
4. Sadilek, A., Kautz, H., Bigham, J. P. : Finding Your Friends and Following Them to Where You Are. In: Fifth ACM International Conference on Web Search and Data Mining (2012)
5. Ye, M., Yin, P., Lee, W.C. : Location Recommendation for Location-based Social Networks. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (2010)
6. Zheng,V.W., Zheng, Z., Xie, X., Yang, Q. : Collaborative Location and Activity Recommendations with GPS History Data. In: WWW '10 Proceedings of the 19th international conference on World wide web, ACM, New York, NY (2010)
7. https://foursquare.com/
8. http://twitter.com/
9. http://mcmc-jags.sourceforge.net/
10. Gelman, A., Rubin, D.B. : Bayesian Data Analysis, Second edition. Chapman & Hall / CRC Texts in Statistical Science (2003)