

# Mining User Mobility Features for Next Place Prediction in Location-based Services

Anastasios Noulas, Salvatore Scellato, Neal Lathia, Cecilia Mascolo  
Computer Laboratory, University of Cambridge  
email: firstname.lastname@cl.cam.ac.uk

**Abstract**—Mobile location-based services are thriving, providing an unprecedented opportunity to collect fine grained spatio-temporal data about the places users visit. This multi-dimensional source of data offers new possibilities to tackle established research problems on human mobility, but it also opens avenues for the development of novel mobile applications and services.

In this work we study the problem of predicting the next venue a mobile user will visit, by exploring the predictive power offered by different facets of user behavior. We first analyze about 35 million check-ins made by about 1 million Foursquare users in over 5 million venues across the globe, spanning a period of five months. We then propose a set of features that aim to capture the factors that may drive users’ movements. Our features exploit information on transitions between types of places, mobility flows between venues, and spatio-temporal characteristics of user check-in patterns. We further extend our study combining all individual features in two supervised learning models, based on linear regression and M5 model trees, resulting in a higher overall prediction accuracy. We find that the supervised methodology based on the combination of multiple features offers the highest levels of prediction accuracy: M5 model trees are able to rank in the top fifty venues one in two user check-ins, amongst thousands of candidate items in the prediction list.

## I. INTRODUCTION

Understanding human mobility has been a long-standing subject in academic research due to the multitude of potential applications. Those range from the better grasp of human behavior and migration patterns, to the evolution of epidemics and spread of disease [5], or the understanding of the mechanisms that shape social networks [7]. With the introduction and increasing popularity of location-based services, the opportunity to study human movement in a qualitatively novel setting is provided. Mobile applications such as Foursquare, where users *check in* broadcasting their visits to places, allow us not only to know the geographic coordinates of a user at a given time, but also the *exact* places they go. A library, a cinema or an airport are a few examples amongst the millions of places which are accessible through these services. Knowledge about the specific places users visit, which goes beyond plain geographic coordinates, can be exploited as an additional dimension to describe human mobility.

In this work we mine user check-in data generated in Foursquare and study the predictive power that different dimensions of the data offer. We formalize the *Next Check-in Problem*, where we aim to predict the exact place a user will visit next given historical data and the current location. The challenge posed in this context is to rank all the potential target places in the prediction scenario, which could easily contain thousands of candidates, so that the actual place visited *next* by the user is ranked as high as possible. This represents a

highly imbalanced prediction scenario, where a single correct instance has to be found (the place a user is going to) amongst thousands of candidate instances.

Having collected approximately 35 million user check-ins over a period of 5 months in 2010, taking place over a set of five million geo-tagged venues, we initially define a set of prediction features that exploit different information dimensions about users’ movements: those include information tailored specifically to an *individual user*, such as historical visits or social ties, and features extracted by mining *global knowledge* about the system such the popularity of places, their geographic distance and user transitions between them. Moreover, we employ a set of features that leverage explicitly *temporal information* about users’ movements. We assess the predictability of individual features and we discover that the most effective features are those which leverage the popularity of target venues and user preferences. Next, we combine the predictive power of individual features in a supervised learning framework. By training two supervised regressors, a regularized linear model and M5 model trees, on past user movements, we demonstrate how a supervised approach can significantly outperform single features in the prediction of future user movements, indicating that user behavior in location-based services is driven by multiple factors who may act synchronously. Notably, M5 Model Trees rank constantly one in two user *check-ins* in the top 50 predicted venues.

## II. MOBILE CHECK-IN DATA

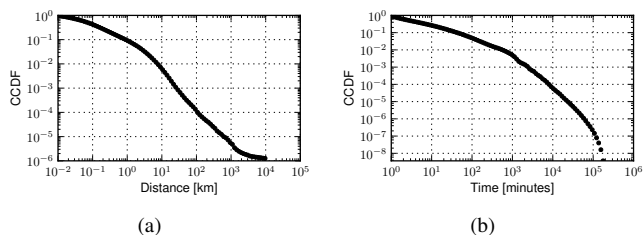


Fig. 1. Complementary Cumulative Distribution Function of (a) spatial distance and (b) time elapsed between consecutive check-ins.

Foursquare is one of the most popular location-based services, with more than 20 million users as of April 2012. The Foursquare mobile application allows users to *check in* venues via their smartphone: they geolocate themselves, broadcast their location to their friends, and participate to the application’s game features. Foursquare users can opt to share their check-ins publicly to their Twitter profiles. We were thus able to crawl for publicly-available check-ins via Twitter’s

streaming API. Note that we can only access those check-ins that users *explicitly* choose to share on Twitter, although users have the possibility to set this option as default for all their check-ins. The resulting dataset contains 35,289,629 timestamped check-ins made by 925,030 users across 4,960,496 venues, over a period of 5 months (May 27th to November 2nd 2010). We estimate that this sample holds approximately 20% to 25% of the entire Foursquare user base at the time of collection<sup>1</sup>. We note that the number of check-ins made by users is highly heterogeneous: the probability distribution exhibits a heavy tail, with about 50% of users having fewer than 10 check-ins. A similar pattern arises when considering the number of check-ins made in each place: only 10% of places have more than 10 check-ins.

The probability distribution of spatial distance between check-ins exhibits a decreasing trend (Figure 1(a)): shorter distances are more likely to appear. However, longer distances are still likely, since the decreasing trend seems to obey an inverse power law  $(\Delta r_0 + r)^{-\beta}$  rather than a faster exponential decay. A power-law fit [3] to the empirical data suggests a decaying exponent  $\beta = 1.50$  and  $\Delta r_0 = 2.87$  km. The distribution of time intervals between consecutive check-ins is shown in Figure 1(b). Longer intervals are less likely than shorter ones, denoting that faster sequences of check-ins might arise, together with long periods of inactivity. This reveals that users exhibit bursts of check-ins that can be mined to understand how they choose where to go next. There are two different trends that become prominent: the first is formed by consecutive check-ins within 1440 minutes (a day) and a second, steeper trend when consecutive check-ins happen across different days. Here, we focus our prediction efforts on check-ins that happen within 24 hours from the previous one.

### III. VENUE PREDICTION IN FOURSQUARE

In this section, we formalize the *Next Check-in Problem*. Given the current check-in of a user, we aim to predict the next place in the city that the user will visit, considering thousands of candidate venues.

#### *The Next Check-in Problem*

We define a set of users  $U$  and a set of locations  $L$ . Each check-in  $c$  by  $u \in U$  is defined as a tuple  $\{l, t\}$ , where  $l \in L$  represents a venue and  $t$  is the check-in's timestamp. The total set of check-ins is denoted as  $C$  and the set of check-ins for a specific user  $u$  as  $C_u$ . We then formalize the next check-in prediction problem as follows. Given a user  $u$  whose current check-in is  $c$  (to venue  $l'$  at time  $t'$ ), our aim is to rank the set of venues  $L$  so that the next venue to be visited by the user will be ranked at the highest possible position in the list. According to the setting described above, the *next check-in* problem is essentially a *ranking task*, where we compute a ranking score  $\hat{r}$  for all venues in  $L$ .

We constrain the selection of candidate venues to the set of places  $L$  within a given city. This approach is justified, if

one bears in mind that almost 99% of consecutive *check-ins* features a distance smaller than 10 kilometers, as shown in Figure 1(a), suggesting that the vast majority of user activity in Foursquare occurs within the urban boundary. Further, it choice allows us to avoid requiring the introduction of distance as an explicit parameter and we can examine its effect as a prediction feature in an unbiased way. Finally, the cardinality of the candidate venue set  $L$  in the prediction list varies from city to city: for the top-33 cities in the dataset we are experimenting with we note that New York features the highest number of venues, 43,681 and Rio de Janeiro trails with 6,788 observed places.

#### *Mobility Prediction Features.*

We now describe in detail the set of prediction features employed to tackle the next check-in problem. For all cases, we note as  $t'$  and  $l'$  the time and location of the current check-in respectively. We set  $t'$  as the current prediction time and we compute the ranking scores of all features assuming knowledge up to that time.

**User Mobility Features** This class refers to features tailored to the check-ins generated by the user under prediction or by her social network. Thus, we aim to capture the likelihood that a user will likely return to a place visited in the past, but also the likes of the user in terms of types of places.

*Historical Visits.* By measuring the number of past visits of user  $u$  at a target venue  $k$ , we are aiming to assess to what extent the next check-in of a user is likely to emerge at a place that has been visited by the user in the past. Formally we have

$$\hat{r}_k(u) = |\{(l, t) \in C_u : t < t' \wedge l = k\}| \quad (1)$$

*Categorical Preferences.* Another source of information based on historical behavior is the number of check-ins user  $u$  has performed at a place that belongs to category  $z$ . In this way, we identify the importance of different categories of places (cinema, coffee shop, football stadium etc.) for a given user and rank them accordingly:

$$\hat{r}_k(u) = |\{(l, t) \in C_u : t < t' \wedge z_l = z_k\}| \quad (2)$$

We note that we subsequently rank venues that belong to the same category by their popularity in terms of check-in number. Thus amongst coffee shops for instance, those with most check-ins are ranked higher.

*Social Filtering.* Considering a user  $u$  and his set of friends  $\Gamma_u$ , we rank a target venue  $k$  by summing the total number of check-ins that any friend  $v$  of the user has performed at place  $k$ :

$$\hat{r}_k(u) = \sum_{v \in \Gamma_u} |\{(l, t) \in C_v : t < t' \wedge l = k\}| \quad (3)$$

**Global Mobility Features.** Now, we demonstrate how we can exploit global information about the check-in patterns of Foursquare users going beyond a specific user and her social network. In this category we will include popularity and geographic features together with features that exploit

<sup>1</sup><http://mashable.com/2010/08/29/foursquare-3-million-users/>

transitions amongst venues.

**Popularity.** We define this feature by counting the total number of check-ins performed by the total set of users  $U$  in the dataset in a venue  $k$ :

$$\hat{r}_k(U) = \sum_{u \in U} |\{(l, t) \in C_u : t < t' \wedge l = k\}| \quad (4)$$

**Geographic Distance.** To study the effect of geographic distance in location-based social services we consider the current location  $l'$  of user  $u$  we measure the distance  $dist(l', k)$  to all other places based on their geographic coordinates. Venues are subsequently ranked in ascending order.

$$\hat{r}_k(l') = dist(l', k) \quad (5)$$

**Rank Distance.** Similarly to geographic distance, we define *rank distance* which measures the relative density between the current place of the user,  $l'$ , and all other places. Formally, considering all places  $l \in L$  we define

$$\hat{r}_k(l') = |\{l \in L : dist(l', w) < dist(l', k)\}| \quad (6)$$

which in plain words translates to the enumeration of venues that are geographically closer to  $l'$  than the destination  $k$ . Our assumption here is that the movement of people is not based on absolute distance values, but rather by the density of opportunities or resources nearby.

**Activity Transitions.** By assuming that the succession of human activities is not random, as for instance we may visit the supermarket after work or go to a hotel after landing at an airport, we are defining the corresponding feature which enables us to capture this signal in Foursquare check-in data. Formally, by noting as a tuple,  $(m, n)$ , the places  $m \in L$  and  $n \in L$  involved in two consecutive check-ins, with  $z_m$  and  $z_n$  being their corresponding categories, we have

$$\hat{r}_k(l') = |\{(m, n) \in L_c : z_m = z_{l'} \wedge z_n = z_k\}| \quad (7)$$

where  $L_c$  denotes the set of tuples for places involved in consecutive transitions before current prediction time  $t'$ .

**Place Transitions.** By definition of the *next check-in problem* we seek to predict consecutive transitions of users across venues. Thus, we build a feature that directly exploits this information, by measuring the direct transitions between all pairs of venues in the city. Accordingly, the rank score of a target venue  $k$  is obtained by enumerating the *past* transitions observed by any user from the current location  $l'$  to location  $k$ , which we formally define as

$$\hat{r}_k(l') = |\{(m, n) \in L_c : m = l' \wedge n = k\}| \quad (8)$$

**Temporal Features.** Here, we define time aware features that capture information both on user activity in terms of visiting categories of places, but also temporal patterns of visits to specific places. More specifically, given that  $z_k$  denotes the type of the target place  $k$ , we define the *Category Hour*

Feature	APR	ACC@10	ACC@50
Random Baseline	0.5	0.0001	0.0005
<b>User Mobility</b>			
Historical Visits	0.68	0.30	0.36
Categorical Preference	0.84	0.006	0.05
Social Filtering	0.61	0.17	0.24
<b>Global Mobility</b>			
Place Popularity	0.86	0.07	0.16
Geographic Distance	0.78	0.08	0.19
Rank Distance	0.78	0.08	0.19
Activity Transition	0.60	0.03	0.06
Place Transition	0.60	0.17	0.20
<b>Temporal</b>			
Category Hour	0.56	0.01	0.02
Category Day	0.57	0.01	0.03
Place Day	0.76	0.07	0.16
Place Hour	0.79	0.09	0.20

TABLE I  
MOBILITY FEATURE APR, ACCURACY@10 AND ACCURACY@50.

popularity as the sum of past check-ins at a place of type  $z_k$  in a given hour  $h$  of the day.

$$\hat{r}_k(t') = |\{(l, t) \in C : z_l = z_k \wedge tod(t) = tod(t')\}| \quad (9)$$

where  $tod(t) \in [0, 1 \dots 24]$  returns a value corresponding to the hour of the day of time  $t$ . Similarly, we set *Category Day* popularity as the sum of check-ins at a place of type  $z$  at a given hour of a week:

$$\hat{r}_k(t') = |\{(l, t) \in C : z_l = z_k \wedge tow(t) = tod(t')\}| \quad (10)$$

where  $tow(t) \in [0, 1 \dots 167]$  returns a value corresponding to the *hour* of the week of time  $t$ .

Finally, we also define the temporal check-in activity at specific venues. We measure the number of check-ins place  $k$  has during a day of the week (*Place Day*) defined as:

$$\hat{r}_k(t') = |\{(l, t) \in C : l = k \wedge dow(t) = dow(t')\}| \quad (11)$$

where  $dow(t)$  returns the day of the week of time  $t$ . A similar definition follows for the number of check-ins that place  $k$  has at a given hour of a day (*Place Hour*), aiming to capture weekly and daily patterns, respectively:

$$\hat{r}_k(t') = |\{(l, t) \in C : l = k \wedge tod(t) = tod(t')\}| \quad (12)$$

#### IV. EVALUATING MOBILITY FEATURES

##### *Methodology and Metrics*

Given each user check-in eligible for prediction, we have a set  $L$  of candidate places to rank. The features compute a numeric value  $\hat{r}_k$  for each candidate venue  $k$ , which are subsequently used to produce a personalized ranking of the venues. We then denote with  $rank(k)$  the rank of venue  $k$ , obtained after sorting in decreasing order all venues in  $L$  according to  $\hat{r}_k$ . We aim to measure the extent that the future venue that will be visited is highly ranked by the prediction algorithms. We use two metrics to measure performance. First, the *Percentile Rank* [6] ( $PR$ ) of the visited place  $k$ :  $PR = \frac{|L| - rank(k) + 1}{|L|}$ . The  $PR$  score is equal to 1 when the place that will be visited next is ranked first and it linearly decreases to 0 as the correct place is demoted down the list.

The *Average Percentile Rank (APR)* is obtained by averaging across all user check-in predictions: this measure captures the average normalized position of the correct instance in the ranked list of instances. We also use *prediction accuracy* to assess the performance when using different prediction list sizes  $N$ . In this case, we successfully predict the next check-in venue if we rank a venue in the top- $N$  places. Average accuracy is the fraction of successful instances over the total number of prediction tasks, which we note as *Accuracy@N*.

### Feature based venue prediction

**APR Results:** The APR results for all features are presented in Table I. From the class of **User Mobility** features, we can distinguish the *Categorical Preference* feature which achieves a score 0.84, which is considerably higher than the *Historical Visits* ( $APR = 0.68$ ) and *Social Filtering* ( $APR = 0.61$ ). This provides an indication that the types of places users tend to visit (cinema, nightclub, coffee shops etc.) can be highly informative about user mobility preferences and could be employed in mobile applications such as place recommendation systems. With respect to features mined exploiting **Global Mobility** patterns of Foursquare users, *Place Popularity* which ranks venues according to the number of past check-ins is the most promising predictor with an  $APR$  score that averages 0.86. The *Geographic Distance* and *Rank Distance* attain an average score 0.78, highlighting that spatial distance is an important factor in the way users decide which venue to visit next. Continuing in the same class of features, the *Activity Transition* and *Place Transition* features achieve lower scores with  $APR = 0.60$ , remaining though higher than the *Random Baseline* which would achieve 0.50. We close the  $APR$  score analysis by looking at the performance of features that exploit **Temporal Information** about the check-in patterns of Foursquare users. The *Place Hour* feature, which ranks target venues according to the frequency of visits by any user observed in the past at the current check-in *hour*, achieves the highest score, 0.79. The *Place Day* ranking, which instead ranks venues by the past number of visits at the *day* of the current user check-in, follows closely with an  $APR = 0.76$ , perhaps due to its lower temporal specificity (day of week instead of hour of day). Nonetheless, both features signify that temporal activity around venues constitutes a source of high quality signal in the venue prediction task. Finally the *Category Hour* and *Category Day* features trail in performance with scores 0.56 and 0.57 respectively.

**The Effect of Prediction List Size:** The  $APR$  scores denote how well, in general, a prediction feature ranks the next visited venue amongst all candidate venues  $L$ . However, in the context of a real mobile application where a finite set of places may be recommended to a user, due to interface or other constraints, one would be interested to examine how prediction approaches perform when the size of the prediction list  $N$  is limited. We have evaluated all algorithms across various top- $N$  lengths using the  $Accuracy@N$  metric. We show the full set of results in Figure 2 and we report the results of  $Accuracy@10$  and  $Accuracy@50$  in Table I. The principal observation is that

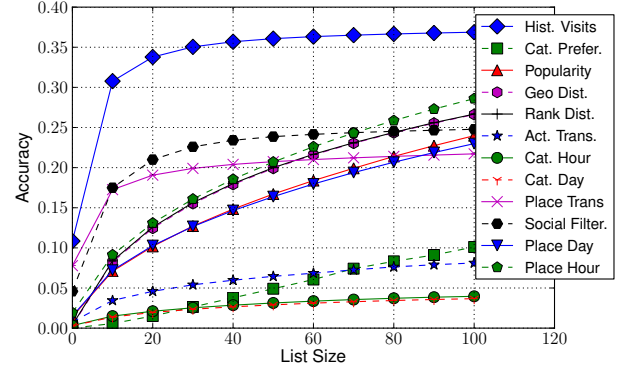


Fig. 2. Feature Predictability: Mean Accuracy for all features when they are being tested on an individual basis for different prediction list sizes  $N$ .

features who rank low in  $APR$  can potentially demonstrate good performance in accuracy terms, in contrast to the results presented in the previous paragraph. Overall, the results in Figure 2 suggest features tailored specifically to **User Mobility** patterns, such as *Historical Visits* and *Social Filtering* dominate in accuracy for list sizes smaller than  $N = 60$ . In particular, *Historical Visits* persist over larger list sizes, up to  $N = 100$ . We note that both features had relatively low  $APR$  scores. On the other hand, features that harvest upon **Global Mobility** information, such as *Place Popularity* or *Geographic Distance* fail to achieve high accuracy scores for small  $N$  values. This *duality* in the performance of the various predictors can be explained by the fact that some features can predict *exactly* the next place a user is going to when, for instance, the user returns to a previously visited place or visits places that their friends go to. Nevertheless, the same features fail to rank appropriately the thousands of previously unseen venues in the city, thus exhibiting a low  $APR$  score. We shall see that those heterogeneities in feature performance will be dissected when we will combine them in a supervised framework.

**Predictability Over Time:** We have demonstrated the overall performance of various features in light of two different metrics,  $APR$  and  $Accuracy@N$ . Another interesting aspect to consider is how well the different prediction strategies may perform at different temporal instants. Figure 3 compares the performance of the various features by showing the temporal evolution of the  $APR$  score over the week. Overall, the effectiveness of each feature over time changes: predictions are more accurate at noon and less accurate in the evening. This suggests that people might be more habitual during the day and more likely to alter their patterns and try something new in the evenings. Interestingly, in the cases of *Geographic Distance* and *Rank Distance* performance is inverted: users are more likely to cover shorter distances at night between consecutive check-ins. Further, the variance between the minima and maxima in the temporal results is more prominent for some features. More specifically, algorithms such as *Historical Visits* and *Place Transition* drop significantly over weekends, whereas *Categorical Preference*, *Place Popularity* and the distance based features are more stable.

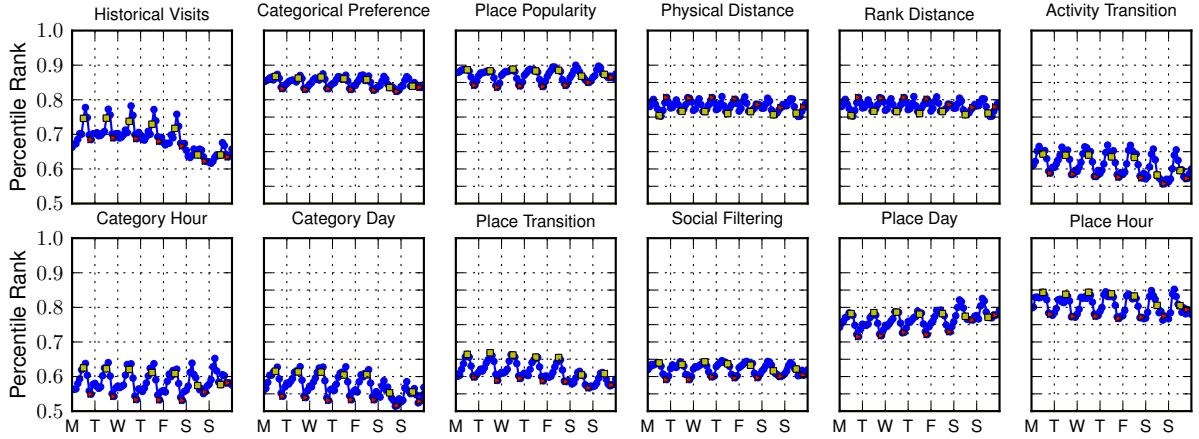


Fig. 3. Feature Weekly Predictability: Average Percentile Rank for all features for different hours of a week. Strong daily periodicities are also observed: notice the yellow circles and red squares which correspond to noon and dinner times respectively.

## V. SUPERVISED LEARNING FOR VENUE PREDICTION

In this section, we combine each of the individual prediction features presented previously into a supervised learning framework. Our aim is to exploit the union of individual features in order to improve predictions, assuming that user mobility in Foursquare is driven by multitude of factors acting synchronously. To predict the next check-in venue of a user we train supervised models assuming knowledge up to prediction time  $t'$ . For every check-in that took place before  $t'$ , we build a training example  $\mathbf{x}$  which encodes the values of the features of the visited venue (e.g., popularity, distance from previous venue, temporal activity scores) and whose label  $y$  is positive. Then, we retrieve a negative labeled input by sampling at random across all other places in the city. Essentially, we are aiming to teach the model what the crucial characteristics are that would allow to differentiate places that attract user check-ins from those which would not. This method of training a model by providing feedback in the form of user preference has been established in the past [4] and corresponds to an effective reduction of the *ranking problem* to a *binary classification task*. Finally, we consider two different supervised models to learn how feature vectors  $\mathbf{x}$  correspond to positive and negative labels: linear ridge regression and M5 decision trees [8].

**Results:** We are now presenting the prediction results obtained when we train and test the two supervised learning models. The M5 trees have the best performance across all models, with an APR of 0.94 and a clear margin compared to all single feature prediction strategies that achieve at best 0.86 when venues are ranked according to *Place Popularity*. On the other hand, the linear regression model achieves an APR score equal to 0.81 which ranks it lower than the popularity and categorical preference features.

If we consider the performance of the models in terms of prediction accuracy (see Figure 4), we can notice that M5 model trees dominate with *Accuracy@10* equal to 0.31

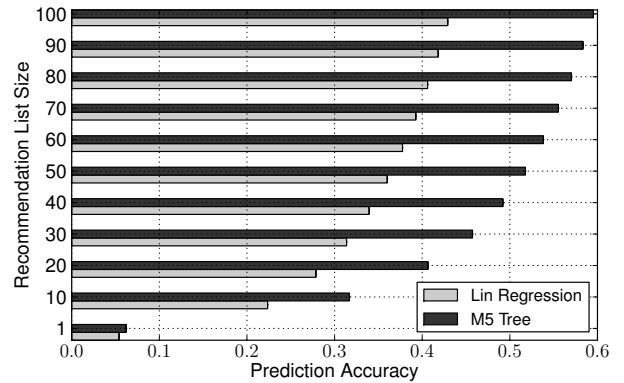


Fig. 4. Average accuracy for the supervised learning algorithms (linear regression and M5 model trees) for different recommendation list sizes.

and *Accuracy@50* equal to 0.51. In the latter case, the next place visited by the user is on average ranked at the top-50 positions of the prediction list, which is a remarkable performance if one considers the multitude of places being ranked in a city. Compared to the *Historical Visits* feature that does best in terms of accuracy, M5 model trees present constantly better performance: *Historical Visits* offer good accuracy scores which however reach an upper bound when prediction list size  $N = 10$ , whereas for larger  $N$  values no improvement is observed. As the reader may notice by inspecting Figure 4, M5 model trees accuracy performance ceases to increase rapidly only when  $N = 100$ . That means that their predictive power is not biased by a small set of candidate venues as in the cases of *Historical Visits* and *Social Filtering*. The linear model presents similar trends in terms of how its accuracy scores improve relative to list size  $N$  but it fails to achieve high absolute scores, although it still does better than *Historical Visits* for  $N$  bigger than 50. Overall, M5 model trees attain peak performance both in *APR* and *Accuracy* terms, showing not only that a supervised approach that combines multiple features is more effective, but also the fact that this combination is more effective in a

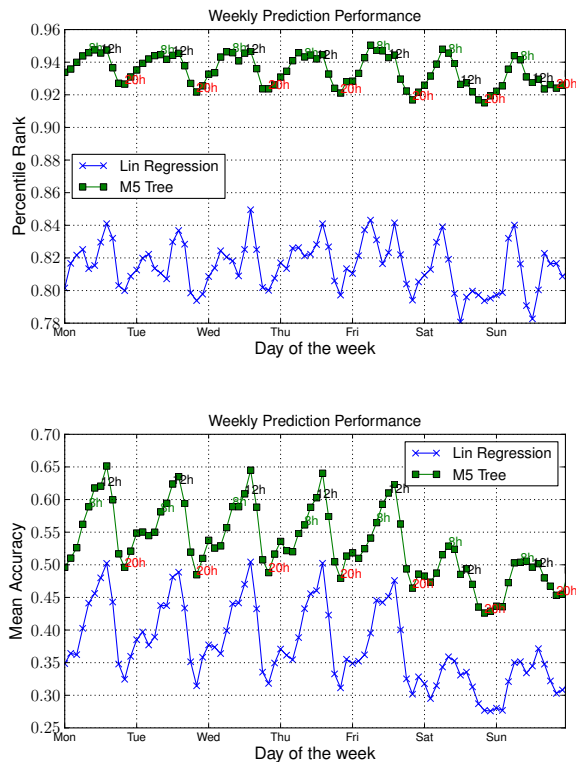


Fig. 5. Temporal evolution of APR (a) and accuracy (b) for the supervised learning algorithms.

non-linear embedding.

Finally, Figure 5 plots the prediction performance of the combined approaches over the week, both using *APR* (Figure 5(a)) and prediction accuracy (Figure 5(b)). Model trees excel in terms of prediction accuracy (shown here for  $N = 50$ , with all  $N$  shown in Figure 4), scoring above 0.5 in general, denoting that one in two user check-ins are successfully predicted. The evolution of temporal predictability presents similar patterns to those observed for individual algorithms. At morning and noon prediction accuracy nears 0.65, whereas over the night the performance drops almost by 25%. Notably, accuracy also drops during weekends, as the *Historical Preferences* and *Place Transitions* algorithms who are also scoring high in this metric also did (see Figure 3). This signifies that the predictability of user movements may decrease at given times, perhaps when they more likely to deviate from their regular mobility patterns by making more randomized choice of places.

## VI. RELATED WORK

Even though location-based services have only recently enjoyed mainstream popularity, they have already attracted research efforts thanks to the new wealth of social and spatial data they offer [1]. In particular, the additional information coming from the places visited by users has been successfully used to improve social link prediction systems [10]. A recent work presents a mobility model that combines social and

spatial factors to reproduce user movements [2]. The main difference between this work and our approach is that we focus on the places visited by users to extract features, while the statistical model in [2] ignores places and does not offer insights on the importance of different factors as space and time vary. A different approach that exploits social networking information to infer the current location of a user has been proposed in [9]. In this case the authors propose a supervised learning model based on the places visited by a user’s friends and they test via cross validation. Instead, our supervised models consider a much larger set of candidate places and, thanks to our longitudinal data, we train them in a more realistic prediction setting on past check-ins, testing on future movements. This also makes our approach suitable for prediction on new users with few or zero check-ins or friends, thanks to features that make no use of historic user activity such as popularity and distance.

## VII. CONCLUSION

This work has addressed the problem of predicting the next place users of location-based services visit. We have collected and analysed a large-scale dataset of user check-ins from Foursquare, and we have defined and evaluated a number of prediction strategies that range from users’ historical visits to features mining global information about the system such as the geographic distance between places, their popularity or temporal patterns of visiting preferences. Our results demonstrate that approaches based only on a particular facet of user behavioral data fail to achieve peak performance. We have proposed and evaluated an approach that combines different mobility features and treats the next place prediction task as a supervised learning problem, with the aim to simultaneously exploit the predictive power of multiple factors. We evaluated two distinct supervised classifiers: a linear regression model and M5 model trees. The latter achieves noticeably higher performance than any other individual feature approach, improving prediction scores across different evaluation metrics.

## REFERENCES

- [1] Z. Cheng, J. Caverlee, K. Lee, and D. Z. Sui. Exploring millions of footprints in location sharing services. In *ICWSM '11*, 2011.
- [2] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *KDD '11*, 2011.
- [3] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-Law Distributions in Empirical Data. *SIAM Review*, 51(4), 2009.
- [4] W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10(1):243–270, May 1999.
- [5] V. Colizza, A. Barrat, M. Barthelemy, A.-J. Valleron, and A. Vespignani. Modeling the worldwide spread of pandemic influenza: Baseline case and containment interventions. *PLoS Med*, 4(1):e13, 01 2007.
- [6] Y. Hu, Y. Koren, and C. Volinsky. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of ICDM '08*, pages 263–272, 2008.
- [7] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins. Geographic routing in social networks. *PNAS*, 102(33):11623–11628, 2005.
- [8] J. Quinlan. Learning with continuous classes. In *AI '92*, 1992.
- [9] A. Sadilek, H. Kautz, and J. P. Bigham. Finding your friends and following them to where you are. In *Proceedings of WSDM '12*, 2012.
- [10] S. Scellato, A. Noulas, and C. Mascolo. Exploiting place features in link prediction on location-based social networks. In *KDD '11*, 2011.