Analysis and Applications of Smartphone User Mobility

Swati Rallapalli, Wei Dong, Gene Moo Lee, Yi-Chao Chen, Lili Qiu
The University of Texas at Austin

***Abstract*—Users around the world have embraced new generation of mobile devices such as the smartphones at a remarkable rate. These devices are equipped with powerful communication and computation capabilities and they enable a wide range of exciting location-based services, *e.g.*, location based ads, content prefetching etc. Many of these services can benefit from a better understanding of the smartphone user mobility, which may differ significantly from the general user mobility. Hence, previous works on understanding user mobility models and predicting user mobility may not directly apply to smartphone users. To overcome this, in this paper we analyze data from two popular location based social networks, where the users are real smartphone users and the places they check-in represent the typical locations where they use their smartphone applications. Specifically, we analyze how individual users move across different locations. We identify several factors that affect user mobility and their relative significance. We then leverage these factors to perform individual mobility prediction. We further show that our mobility prediction yields significant benefit to two important location based applications: content prefetching and shared ride recommendation.**

I. Introduction

Users around the world have embraced smartphones at a remarkable rate [1], [2]. Global smartphone sales in the third quarter of 2011 were up 42% from the third quarter of 2010 [3]. Unlike traditional mobile devices, these new devices have very good computing capabilities and are equipped with various wireless communication interfaces. This enables us to have a wide-range of location-based services, *e.g.*, content prefetching, targeted ads, shared ride recommendation, predicting friendships, generating tour guides. Understanding smartphone user mobility is critical to all of these applications. For example, if we can accurately predict where a user will go next, we can actively prefetch his desired content to that location for a better download experience.

While the importance of mobility analysis has long been recognized, most of the existing research on this subject has been severely limited by the traces available. Most works on mobility modeling and mobility prediction use general traces *e.g.*, Wi-Fi and Cellular [4], [5], [6], [7]. However these works may not directly apply to smartphone users for two reasons. First, the user intent of using the device at specific locations is not captured in these traces. For example, smartphone users may be more likely to use their device at some locations than others, *e.g.*, a train station vs. a meeting room. Second, the demographics of smartphone users maybe different from the general population.

On the other hand, recently Location-Based Social Networks (LBSNs) have experienced an explosive growth in popularity as people around the world have embraced location-sensing mobile devices at a remarkable rate. LBSNs provide a unique opportunity to understand large scale smartphone user mobility. For example, Foursquare [8], the most popular LBSN, had over 10 million users with 1 billion *check-ins* as of September 2011 [9]. There are many other popular LBSN services, such as Gowalla [10], Brightkite [11], and Loopt [12]. Moreover, major social networking sites like Facebook, Twitter, and Google+ have also added location-based features into their services.

In LBSNs, people share their locations with their friends, receive location-based recommendations, and make comments about the places they visited. People record their geographical locations in the form of *check-ins*. If a user wants to check-in, she uploads her geographic coordinates to the server. Then the server gives back a list of possible places and lets the user select the location through a check-in or create a new place if the location has not yet been registered at the server. They differ from existing mobility data in that: (i) the check-in volume is massive, (ii) a check-in gives more fine-grained location information than inference based on the cell tower signals, (iii) each check-in not only contains geographic location but also includes semantics behind the location (restaurants, offices, shops, etc.), (iv) a check-in does not include continuous movements but a point location, and users check in only if they want to, so there is explicit user intention to share the location, (v) in addition to check-ins, we have on-line friendship information among users.

We collect and analyze the data from two major LBSNs: Foursquare and Gowalla. We first pick factors that help predict human mobility, and then identify their relative significance. We then leverage these factors to perform mobility prediction. We further show that our mobility prediction can potentially help prefetching to increase the hit rate by 0.12x - 33x than passive caching and help recommend shared rides in 10% - 100% of time with only 0%-26% false positives.

II. Data Collection

We collect and analyze the data from Foursquare and Gowalla, which are two of the most popular LBSNs. As of October 2011, Foursquare has 14 M users and Gowalla has 400K users. We collect the traces using the open APIs provided by Foursquare [13] and Gowalla [14]. The Foursquare API does not allow us to retrieve check-in history of individual users, but they provide a list of visitors for a given venue in a two-hour window. In order to get the user trajectory, we pick popular venues and periodically fetch the recent visitors to those places every two hours. We may miss check-in information at unpopular places. However, we can still get good coverage of people's movements across popular venues. Gowalla allows us to directly crawl all check-ins for a given user, so we have check-in information at both popular and unpopular venues.

We collect user and venue information. The user information includes user's name, home city, friends, tips on venues and venue information includes venue's name, latitude and longitude, categories (*e.g.*, Airport, College & University, Food, etc.), and numbers of visitors and check-ins. We collected $277,900$ users across $13,484$ venues from 01/02/2012 to 02/06/2012 from Foursquare, and $51,363$ users across $66,578$ venues from January 2009 to December 2011 from Gowalla. We analyze $5,288$ active users with $92,985$ check-ins at popular venues in Foursquare and $1,442$ active users with $9,896$ check-ins at popular venues in Gowalla. Active users are those who have at least 15 check-ins in a 5-week period, and popular venues are those that have at least 25 check-ins in a 2-week period, starting from 01/02/2012 in Foursquare and from 05/01/2010 in Gowalla.

III. Mobility Prediction

Predicting users' check-ins has many important applications, such as targeted ads, content prefetching, shared-ride recommendation, planning friends' hang-out. In this section, we develop a scheme to predict a user's next check-in.

A. Predicting next check-in

We define user transition matrix $M(u,t,a,b)$ as user u's transition probability from location a to location b in a time interval t, which could be few hours, a day, a week, etc. as defined by us. We specify the length of t used for our analysis below. It can be easily computed as the ratio of the total number of transitions from a to b divided by the total number of transitions out of a. By varying the granularity of time interval and locations, we can get a range of different levels of transition matrices.

We apply different levels of transition matrices for mobility prediction. In particular, mobility pattern varies across users and across time. Suppose we want to predict user u's next check-in given its current check-in venue v_j at time t. At one end of the spectrum, we can use the same user's previous trajectory taken around the same time to compute transition probability. This gives the most specific information, but may yield too few data samples to meaningfully compute the transition probability. At the other end of spectrum, we could take all users' previous trajectories across all the time to derive the transition probability. This gives us broadest coverage, but may be too general and does not fully capture the user or time specific information. There are many levels in between. We first identify the following *metrics* that we leverage for mobility prediction:

1. User-venue-time specific: We only consider the transitions for the same user from the same venue in the same time bin when computing the transition probability. That is, we compute the transition probability for user u to transit from venue v_j to venue v_k in time bin t, denoted as $P[u, v_j, v_k, t]$, as the ratio between the total number of such transitions and the total number of visits of user u to venue v_j in time bin t. We partition a day into six time bins: (i) 0 to 6am, (ii) 6 to 10am, (iii) 10am to noon, (iv) noon to 5pm, (v) 5 to 10pm, (vi) 10pm to 0am. Different time-bin sizes are used according to changes in users' mobility patterns.

2. User-venue specific: Similar to the above, except that we only consider visits other than those that occurred in time bin t. We exclude these visits to avoid double counting the visits that occurred in time bin t.

3. Friend-venue-time specific: Similar to the first metric, we now consider user u's friends instead of user u, where friends are obtained using the friendship information of the user published at the LBSNs. We do not consider user u's visits to avoid double counting.

4. Friend-venue specific: Similar to friend-venue-time specific, but now we relax the time constraint by considering all time other than time bin t.

5. Venue-time specific: Similar to user-venue-time specific, except that we now consider all users other than u and u's friends. Again user u and u's friends are excluded to avoid double counting.

6. Venue global: We compute the transition probability from v_j to v_k by considering all users across all the time except user u, u's friends, and time bin t. That is, the fraction of transitions from venue v_j are to venue v_k, conditioning on the transitions that are made by anyone other than user u, u's friends, or in time bin t.

7. User global: We compute the transition probability based on how often user u visits venue v_k (regardless of time and the previous venue). That is, the fraction of user u's visits that are to venue v_k.

8. Friend global: We compute the transition probability based on how often user u's friends visit venue v_k (regardless of time and the previous venue).

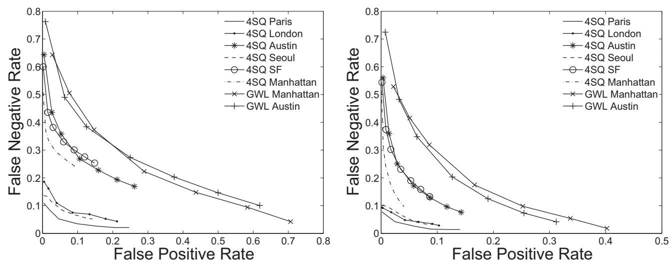
Each of these metrics is intuitively useful, but how to combine them to make a a good predictor is a challenge. To address this, we apply regression to automatically learn the importance of each metric based on the training traces collected from the past. Specifically, we cast the weight estimation into a regression problem: $Ax = b$, where $A(i,j)$ denotes the transition probability into venue v_i according to the j-th metric, $x(j)$ denotes the weight of the j-th metric (which is one of the above eight metrics), and $b(i)$ is a binary indicator of whether the next check-in is at venue v_i. From the previous traces, we get A and b. We estimate x by finding the closest solution that matches $Ax = b$. The estimated x indicates that the above rules (1) and (7) are most important. We use the estimated x and all the metrics for our evaluation. Then we make predictions of future check-ins by constructing A from the traces seen so far and applying the estimated x to compute b. We rank the predicted venues in a decreasing order of b, and pick the top $K\%$ venues as predicted next check-ins. We vary K (between 1 and 100) to trade-off between *false positive (FP)* and *false negative (FN)*, where FP and FN are computed as follows:

$$FP = \frac{\text{\# incorrectly predicted next check-ins}}{\text{\# venues not next check-in}} \tag{1}$$

$$FN = \frac{\text{\# next check-ins missed}}{\text{\# next check-in}} \tag{2}$$

We use week 1 for bootstrap to accumulate enough check-in history. Then we apply regression to learn weight x by extracting $Ax = b$ using weeks 2 and 3 as training traces. Then we apply the estimated x to predict next check-ins for testing traces beginning from week 4. We use Foursquare trace from 01/02/2012 to 02/06/2012 and Gowalla trace from 05/01/2010 to 06/09/2010. We filter out users and venues with too few check-ins as explained in Section II. Figure 1 (a) shows False Positive (FP) Rate versus False Negative Rate (FN). We make two important observations. First, our prediction scheme achieves good accuracy: 2% - 25% false negatives with 9% -

66% false positives. Second, the accuracy varies a lot across cities. For example, in Foursquare Paris we achieve a FN Rate of as low as 11% at 0.3% FP Rate, whereas in Foursquare Austin, we see a FN Rate of 64% at FP Rate of 0.4%. This is due to varying check-in patterns of the users in cities. In Paris, users tend to check-in at a venue regularly (*e.g.*, train stations) and the accuracy is high. In Austin, which does not have good public transportation and popular locations are spread across the town, the check-in pattern is harder to predict. Other cities are in between.



(a) Next check-in venue (b) Next check-in category

Fig. 1. Accuracy of next check-in prediction.

B. Predicting next check-in category

Some applications like targeted ads only require us to predict the categories of venues that will be checked in next. For instance, if a user is going to Italian restaurant next, sending him ads about any Italian restaurant in the area is useful (the ads he sees and the discounts may affect his choice of the restaurant, if he is already not too keen on a specific one). We can simply apply the above prediction scheme, and then aggregate the predicted venues to predicted categories (*i.e.*, the probability of visiting a category is the probability of visiting any venue that falls into the category). We again evaluate the FP rate and FN rate by varying the top $K\%$ categories. Figure 1 (b) summarizes the results. As before, we see varying accuracy of prediction across different cities. For example, Foursquare Paris has a FN Rate of 7.8% and FP Rate of 0.2%, and Foursquare Austin has FN Rate of 67% and FP Rate of 0.7%. Moreover, we can achieve a coverage of 87% to 98.6%, higher than in case of predicting next check-in, since we are looking for more coarse-grained information.

IV. Applications

Mobility prediction has many applications. We focus on content prefetching and shared ride recommendation.

A. Content Prefetching

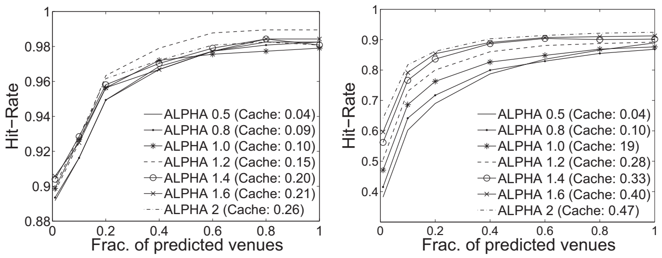
If we know a user's next check-in, we can prefetch content in advance so that by the time the user arrives, the content is already available locally at the cellular base station or Wi-Fi access points and the user can enjoy the much higher local wireless capacity instead of being bottlenecked by the slow Internet link. This is especially useful when the gap between the access link capacity and local wireless capacity is large. For example, 802.11n can give speeds up to 600 Mbps [15], while the access link capacity is often on the order of a few Mbps. The gap tends to further increase over time due to recent advances in wireless technology compared with the much slower deployment/upgrade in access link capacity.

To evaluate the performance of prefetching, we assume a user has a new content request at all his check-ins. We generate the content demand using a Zipf-like distribution (*i.e.*, the number of requests for the i-th most popular file is proportional to $1/i^{\alpha}$, where α is a small constant) [16], [17]. It determines the skewness of the demands. The higher the α, the more skewed the demand is. We assume that the content is made available at predicted venues instantaneously when the prediction is made. Any user who visits the location after that can download the file until it is purged out. We compare against a pure caching scheme, where the content is cached at the venue only *after* it is viewed for the first time. For both schemes, we assume a cache of 1 TB [18] (since storage is getting inexpensive) and file size of 5 GB (typical movie size, to be safe we pick a big file size, when video file sizes are smaller, hit-rates should be even better due to less frequent purging out) and replace using the *Least Frequently* used file when the cache is full. We quantify the hit-rate, which is the fraction of requests that can be served from the local cache.

Figure 2 (a) shows the hit-rates for Foursquare Paris and figure 2 (b) shows the hit-rates for Foursquare Austin. We pick these two cities since they represent two extreme cases in terms of mobility prediction accuracy. The results from other cities show similar trend and are omitted for brevity. In both the figures, X-axis shows varying fraction of the predicted venues we prefetch the content to and Y-axis shows the hit-rate. A higher hit-rate means more requests are being served from the local cache and is preferred. The hit-rate for passive caching per α in a city is shown in brackets in the legend. It's only one number because it doesn't change with the fraction of venues. We first observe that we prefetch the content to a higher fraction of the predicted venues, hit-rate is better because there is a higher probability that the actual check-in venue falls within the set of venues we prefetch the content to. We also observe that prefetching consistently out-performs passive caching. The amount of improvement depends on the value of α, for Foursquare Austin, when $\alpha = 2$, the hit rate of both prefetching and caching are high: 64% and 47%, respectively. Decreasing α enlarges the gap between the two. For example, when $\alpha = 1$, the hit rates of prefetching and caching are 47% and 19%, respectively; when $\alpha = 0.5$, the corresponding numbers become 38% and 4%, respectively. That's because an increased α indicates popular files get more requests (higher skewness) and the cost of fetching it for the first time is amortized by many future requests. Further 3 shows the hit-rate for all cities when alpha = 1. We observe that prefetching helps improve hit-rate even for cities that have lower accuracy of check-in prediction, because the prefetched content can be used to serve someone else even if the target user does not check-in as predicted. In general, the level of our mobility prediction accuracy is sufficient to significantly benefit prefetching.

B. Recommending Shared Rides

Mobility prediction is also useful for finding shared rides. Existing systems (*e.g.*, [19]) require both riders and ride givers to provide their source and destination locations and the time of travel, based on which they perform match making. This approach may easily miss potential riding opportunities as users may not always keep their information up to date. In comparison, recommending shared rides based on mobility



(a) 4SQ, Paris (b) 4SQ, Austin

Fig. 2. Hit-Rate for caching and prefetching.

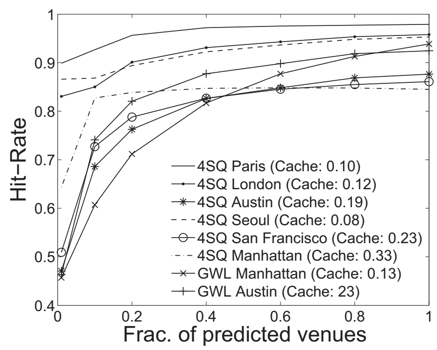
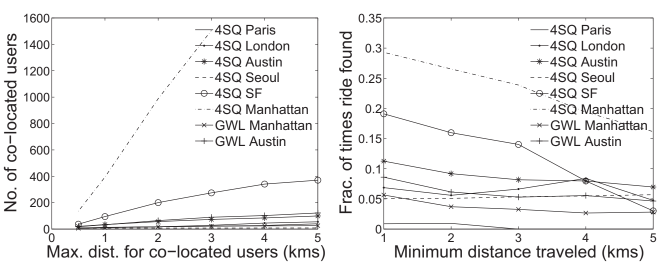


Fig. 3. Hit-Rate for caching and prefetching (α = 1).

prediction eliminates the need of user input and can automatically recommend ride based on their current locations and predicted future check-ins. The requester can contact the recommended ride givers to confirm the ride.

First, we examine how often we find shared rides. We define a possible shared ride as follows: (i) the current locations of the two users are within d_c km of each other, (ii) their next check-ins are at least d_a away from the current locations (*i.e.*, not walkable), (iii) their next check-in locations are within d_c km, and (iv) the next check-ins of the two users occurred within 24 hours of each other.



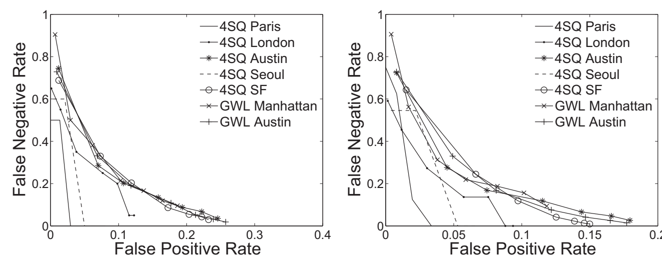
(a) Co-located users (b) Fraction of times shared-ride found

Fig. 4. Average number of co-located users, every person finds.

Figure 4(a) shows the average number of co-located users *i.e.*, the number of people who are grouped to be eligible for shared-ride, as we vary d_c. Figure 4(b) shows the fraction of times that a user finds a shared ride (out of the number of times that he travels farther than d_a kms between consecutive check-ins), where d_c is fixed at 1 km and we vary d_a. As we would expect, in more popular cities, there are more co-located users and it is more likely to find shared rides. For example, in Foursquare Manhattan and San Francisco users find shared rides 24% and 14% of the times they travel more than 3 kms; in comparison, in Foursquare Austin and Seoul, users find shared rides only 8% and 5% of the times.

Next we examine how accurately we can recommend shared rides based on our mobility prediction. Specifically, given a requester's current location and (true) next check-in location, where the distance between two check-ins are at least d_a away and not walkable, we want to recommend ride givers who are currently within d_c km from the requester and have (predicted) next check-ins within d_c km from the requester's next check-in (within 24 hours). For this evaluation since we do not have the meta data to show who is the rider and who is the ride-giver, we assume both the cases: where each person becomes the rider and ride-giver, and the results are average of both.

Figure 5 shows the FN Rate versus the FP Rate for different sets of co-location distance d_c and minimum distance for shared-ride d_a. The accuracy of ride-recommendation is generally high in all cases. London has high mobility prediction accuracy, so its accuracy of ride recommendation is also high: 14% false negatives at 6% false positives. San Francisco has lower mobility prediction accuracy, so its accuracy of ride recommendation is lower but still pretty good: 24% false negatives at 7% false positives when $d_c = 2$ kms and $d_a = 2$ kms. Moreover, we see that the recommendation accuracy follows similar trend for a different d_c and d_a.



(a) d_c = 1km, d_a = 1km (b) d_c = 2km, d_a = 2km

Fig. 5. Accuracy of ride recommendation.

C. Discussion

We note that the LBSN check-in data is ideal to understand the performance of smartphone applications, as these are the venues the smartphones users tend to use their applications, as explained in Section I. One downside of using this trace could be – missing the locations where there is no check-in. Due to the voluntary nature of user check-ins, we miss the locations where a user chooses not to check-in (although he maybe willing to use his smartphone at that location for a more compelling application). With increasing usage of smartphones and LBSNs, we believe this bias would reduce. Moreover, capturing more locations would improve the performance of our example applications, *e.g.*, we would be able to find more shared rides and also predict check-ins more accurately due to the availability of richer information.

V. Related Work

Our work is related to: (i) understanding and predicting human mobility and (ii) analysis of location-based social networks.

Understanding and predicting human mobility: There has been significant amount of work on mobility prediction.

Some focus on coarse-grained prediction in cellular networks (*e.g.*, [4], [5], [20], [21]), while others use Wi-Fi records as an indicator of user mobility (*e.g.*, [6], [7], [22]). Different from these works, we focus on smartphone users and use check-in data from location based social networks, which is much larger in scale and is associated with social information and our results are applicable to potential future applications.

Markov models have been widely used in the past for localization and mobility prediction. For example, [23], [24] and several others leverage first order Markov model. Many existing works also leverage a second order Markov model. For example, [22] compares various predictors in the literature and suggests that second order Markov model with a simple fallback mechanism (when there is no prediction) performs well. [6] builds the users' customized mobility models on the devices themselves, and uses a second order Markov model to predict connection opportunities and their quality. Our prediction algorithm also uses a first order Markov model, partly because users do not check-in at all venues, which means intermediary venues might be missing, so a higher order Markov model may not be suitable. Our work differs in that we use LBSN traces, which not only enable us to have the exact location and the semantics of the location but also the user intent in using a smartphone application at that location. Moreover we derive the relative importance of several factors we pick for mobility prediction and show that our example applications can achieve good performance.

There are several interesting works on understanding human mobility. Authors in [25] analyze contact networks by combining data from multiple sources and further generating synthetic data of individuals. Authors in [26], study the mobility of $100,000$ phone users over six-months, conclude that humans travel in simple reproducible patterns, and return to a few highly frequented locations. Authors in [27], analyze human mobility in terms of community behavior, and study inter-community and intra-community contacts separately. The insights from these studies can be leveraged while building the applications we illustrate in this work.

Location-based social networks: Recently, LBSNs have attracted the research community to analyze such massive data [28], [29]. Unlike our work that focuses on user mobility, most works in this area study friendship relationships in LBSNs [30], [31]. [32] is among the few that analyze human mobility. They look at data from Gowalla and Brightkite in 2008 to 2010 along with mobile phone location dataset. It reports that human mobility consists of (1) short-ranged travel that is spatially and temporally periodic (50% - 70%) and (2) long-distance jumps which can be explained by social relationships (10% - 30%).

VI. Conclusion

In this paper, we perform an in-depth analysis of smartphone users' mobility using two of the largest location-based social networks. Our findings suggest that many factors such as time of the day and friends' behavior affect smartphone users' mobility pattern and it's possible to predict user mobility with reasonable accuracy. Our analysis also shed light on microscopic human mobility across different location granularities. These findings have significant implications on the design and evaluation of mobile networks. As examples, we show two applications that benefit from mobility information. As part of our future work, we plan to develop models to capture user mobility at different granularities and explore more mobility-aware applications.

Acknowledgements

This work is supported in part by NSF Grants CNS-0916106, CCF-1117009, and CCF-0916309. We are thankful to the anonymous reviewers for their constructive feedback, to help improve the paper.

References

[1] "Smartphone usage." http://googlemobileads.blogspot.com/2012/01/new-research-global-surge-i%n-smartphone.html.

[2] "Att benefits from smartphone usage." http://online.wsj.com/article/SB100008723963904435709045775466221040677%12.html.

[3] "Increase in smartphone sales." http://ismartdesign.co.nz/explore/blog/item/7-up.

[4] I. F. Akyildiz and W. Wang, "The predictive user mobility profile framework for wireless multimedia networks," *IEEE/ACM Transactions on Networking*, 2004.

[5] A. R. Aljadhai and T. Znati, "Predictive mobility support for QoS provisioning in mobile wireless environments," *IEEE Journal on Selected Areas in Communications*, 2001.

[6] A. J. Nicholson and B. D. Noble, "Breadcrumbs: forecasting mobile connectivity," in *Proc. of ACM MobiCom*, 2008.

[7] L. Song, U. Deshpande, U. C. Kozat, D. Kotz, and R. Jain, "Predictability of WLAN mobility and its effects on bandwidth provisioning," in *Proc. of IEEE INFOCOM*, 2006.

[8] "Foursquare." http://foursquare.com.

[9] "A million check-ins isn't cool. You know what's cool?." http://blog.foursquare.com/2011/09/20/billion/.

[10] "Gowalla." http://gowalla.com.

[11] "Brightkite." http://brightkite.com.

[12] "Loopt." http://loopt.com.

[13] "Foursquare API." http://developer.foursquare.com/docs/.

[14] "Gowalla API." http://gowalla.com/api/docs.

[15] "Wi-Fi Speeds." http://www.broadcom.com/collateral/wp/802_11n-WP100-R.pdf.

[16] P. Mundur, R. Simon, and A. K. Sood, "End-to-end analysis of distributed video-on-demand systems," *IEEE Transactions on Multimedia*, vol. 6, no. 1, pp. 129–141, 2004.

[17] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "YouTube traffic characterization: a view from the edge," in *Proc. of ACM IMC*, 2007.

[18] "Network Storage." http://www.eweek.com/c/a/Data-Storage/Iomega-Offers-1TB-of-Network-Stor%age-for-300/.

[19] "Shared ride." http://erideshare.com.

[20] P. N. Pathirana, A. V. Savkin, and S. Jha, "Mobility modelling and trajectory prediction for cellular networks with mobile base stations," in *Proc. of ACM MobiHoc*, 2003.

[21] T. Liu, P. Bahl, and I. Chlamtac, "Mobility modeling, location tracking, and trajectory prediction in wireless ATM networks," *IEEE Journal on Selected Areas in Communications*, 1998.

[22] L. Song, D. Kotz, R. Jain, and X. He, "Evaluating location predictors with extensive Wi-Fi mobility data," in *Proc. of IEEE INFOCOM*, 2004.

[23] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proc. of ACM MOBICOM*, Sept. 2004.

[24] M. H. T. Martins, H. Chen, and K. Sezaki, "OTMCL: orientation tracking-based monte carlo localization for mobile sensor networks," in *Proc. of the Six International Conference on Networked Sensing Systems (INSS)*, 2009.

[25] C. Barrett, R. Beckman, M. Khan, V. Kumar, M. Marathe, P. Stretz, T. Dutta, and B. Lewis, “Generation and analysis of large synthetic social contact networks,” in *Simulation Conference (WSC), Proceedings of the 2009 Winter*, pp. 1003 –1014, dec. 2009.

[26] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, “Understanding individual human mobility patterns,” *Nature*, vol. 453, pp. 779–782, June 2008.

[27] T. Hossmann, T. Spyropoulos, and F. Legendre, “Putting contacts into context: mobility modeling beyond inter-contact times,” in *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc ’11, (New York, NY, USA), pp. 18:1–18:11, ACM, 2011.

[28] Z. Cheng, J. Caverlee, K. Lee, and D. Z. Sui, “Exploring millions of footprints in location sharing services,” in *Proc. of AAAI ICWSM*, 2011.

[29] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil, “An empirical study of geographic user activity patterns in Foursquare,” in *Proc. of AAAI ICWSM*, 2011.

[30] S. Scellato, A. Noulas, and C. Mascolo, “Exploiting place features in link prediction on location-based social networks,” in *Proc. of ACM KDD*, 2011.

[31] S. Scellato, A. Noulas, R. Lambiotte, and C. Mascolo, “Socio-spatial properties of online location-based social networks,” in *Proc. of AAAI ICWSM*, 2011.

[32] E. Cho, S. A. Myers, and J. Leskovec, “Friendship and mobility: user movement in location-based social networks,” in *Proc. of ACM KDD*, 2011.