Recommending Social Events from Mobile Phone Location Data

Daniele Quercia[§†] Neal Lathia[‡] Francesco Calabrese[†] Giusy Di Lorenzo[†] Jon Crowcroft[§]

[§]University of Cambridge [†]Massachusetts Institute of Technology [‡]University College London

Abstract—**A city offers thousands of social events a day, and it is difficult for dwellers to make choices. The combination of mobile phones and recommender systems can change the way one deals with such abundance. Mobile phones with positioning technology are now widely available, making it easy for people to broadcast their whereabouts; recommender systems can now identify patterns in people's movements in order to, for example, recommend events. To do so, the system relies on having mobile users who share their attendance at a large number of social events: cold-start users, who have no location history, cannot receive recommendations. We set out to address the mobile cold-start problem by answering the following research question: how can social events be recommended to a cold-start user based only on his home location?**

To answer this question, we carry out a study of the relationship between preferences for social events and geography, the first of its kind in a large metropolitan area. We sample location estimations of one million mobile phone users in Greater Boston, combine the sample with social events in the same area, and infer the social events attended by 2,519 residents. Upon this data, we test a variety of algorithms for recommending social events. We find that the most effective algorithm recommends events that are popular among residents of an area. The least effective, instead, recommends events that are geographically close to the area. This last result has interesting implications for location-based services that emphasize recommending nearby events.

*Keywords***-mobile, recommender systems, web 2.0**

I. INTRODUCTION

The ability for mobile phones to provide ubiquitous connectivity and access to information place them amongst the most rapidly growing technologies in the world; in fact, in 2004 mobile phone sales substantially surpassed the number of personal computers sold [7]. The increase of mobile phones equipped with location-sensing technology and enabled to elicit user experiences [8] presents a valuable opportunity: recommender systems can no longer merely support users tackling information overload online, but can now be applied to aide users' decision making in a variety of offline contexts as well. Mobile-web portals, like Rummble.com, already allow users to tag, rate, and describe locations as they visit them, in order to aide the discovery of unknown locations of interest. Similarly, users' location history can be mined in order to provide personalized recommendations for social events as they occur in the surrounding areas.

Liberating recommender systems from web-based scenarios, though, recasts well known problems into new, unexplored domains. How can social events be recommended to a user who has *no location history*? The *cold-start* problem [13] has received ample attention in web recommender systems, but less so for mobile equivalents.

We set out to investigate the possibility of recommending social events to cold-start users based only on input that is indirectly generated when they carry their mobile phones. In particular, we explore the extent that effective recommendations can be computed if users were to only *disclose their home locations*. To do so, we combine two datasets: the first includes location estimates of one million mobile phone users in the Greater Boston area, and the second lists large-scale social events that occurred throughout the same area as the users' locations were logged. By analyzing the two sets of data, we make three main contributions: **(1)** We infer where 80,000 mobile users live and the social events that have been likely attended by 2,519 users (Section II); **(2)** We use the relation between home locations and event attendances to recommend social events using six different algorithms (Section III); **(3)** We evaluate the effectiveness of these algorithms in ranking events (Section IV). Our goal is not algorithmic (we do not propose any new algorithm) but is experimental. We study the individual factors (e.g., geographic distance, event popularity in area of residence) that may explain why people attend certain social events. From our results, we draw the following preliminary conclusions: **(1)** The most effective choice of events is to recommend those that are popular among residents of the area. This result reflects theories that claim that people have a disposition to geographically cluster with others who are like-minded [5]: for the first time, we provide empirical evidence that those theories hold at metropolitan level; **(2)** The least effective way is to recommend events that happen to be geographically close to the area of residence. This suggests that location-based services may currently place too much emphasis on recommending nearby events.

II. INFERENCES FROM MOBILITY AND EVENT DATA

We used two sources of data. The first consists of location estimates of mobile phone users in the Greater Boston area. The second is the list of social events in the same area during the period of our study. By combing the two sources, we are able to determine where city dwellers live and go, and which social events they have likely attended.

Where City Dwellers Live and Go. From *Airsage Inc.* (a company that analyzes mobile location data to offer

traffic congestion solutions), we obtained estimates of the locations of 1 million mobile phone users in the Greater Boston area (20% of the entire population) for one and half months during the summer of 2009. To make our analysis computationally tractable, we sample 80,000 users using a stratified sampling with proportional allocation across zipcodes. We then extract individuals trips (trajectories) for those users and infer where the users stop during the day (as detailed [6]).

Which Social Events They Attend. We crawled the "Boston Globe Calendar" website[1] to extract the social events that took place during our period of study, we divided the $15km^2$ area of Greater Boston in geographic cells of 500 x $500m$, and we placed the events and user stops in the corresponding cells. By intersecting stops and social events across cells, we are able to determine which social events have been attended by whom. To reduce false positives, we apply two filtering rules: 1) we only consider events that are geographically "unique" (no other big event is taking place within the radius of 1 km at the same time); 2) we consider only people who both stay in an event location for at least 70% of the event duration and live somewhere else. These filtering rules reduce our dataset (the number of social events goes from 500 to 53 and the number of mobile users goes from 80,000 to 2,519), but it also makes sure that we will work upon correct inferences with high probability.

Combined Dataset. Our resulting data can be represented as a 2,519 by 53 binary user-event attendance matrix; an entry in position (i, j) of this matrix represents whether user i attended event j. The combined dataset is 97.3% sparse. In other words, there are many user-event pairs that we have no knowledge about (the entries in the matrix are thus zero), and we will see how two algorithms perform on such a sparse matrix (Section IV-C). For now, we only need a means of computing recommendations for cold-start users who have only input their home locations. We therefore clustered the 2,519 users according to their inferred postcode, producing 49 unique groups. We then merged each group's profiles; from these, we are able to construct a 49 x 53 home location-event attendance matrix m. An entry in position (i, j) of this matrix represents how many users from location i attended event j (it is therefore no longer a binary matrix). A key advantage of this reduction is that our matrix is only 50.5% sparse; in the following section, we introduce algorithms to predict attendance and rank events that will operate on this small location-event matrix. Since we know where users live and where the events took place, we also have access to information regarding the *distance* between location i and event j.

III. RECOMMENDING SOCIAL EVENTS

From our analysis of the mobile phone data, we know which social events residents of an area have attended, and we are now able to recommend social events. We do so in six different ways:

(1) Popular events. The simplest way for recommending social events is to recommend the most attended ones in Greater Boston. More formally, $\widehat{score}_{i,j}$, or prediction that users who live in location i will attend event j is proportional to the number of users who have attended event j:

$$\widehat{score}_{i,j} = \sum_i m_{i,j} \qquad (1)$$

For ranking purposes, what matters is not the score itself but how the score compares to another. That is also true for the remaining ways of recommending events.

(2) Geographically close events. An alternative way is to recommend events that are close to the area of residence. In other words, the prediction that users who live in location i will attend event j is inversely proportion to the geographic distance between event j and location i:

$$\widehat{score}_{i,j} = \frac{1}{distance(i, j)} \qquad (2)$$

(3) Popular events in area. We also recommend, to every area's residents, the most popular events among the residents of the area. The prediction is proportional to the number of users who live in location i and have attended event j:

$$\widehat{score}_{i,j} = m_{i,j} \qquad (3)$$

(4) *Term Frequency-Inverse Document Frequency.* One could also identify events that are not necessarily popular in general but are popular among residents of the area. *TF-IDF* (Term Frequency-Inverse Document Frequency) is a widely-used approach in the literature of Information Retrieval [3] that would do just that. To paraphrase this approach in our context, we assign a higher score to social events that are highly popular in a particular location and that may not be necessarily popular in the remaining locations. The assumption is that the more unique an event is for a location, the more representative the event is for that location. Using *TF-IDF* , we learn that each location has its own predominant category. For example, residents of the suburban area of Belmont ("MA 02478") tend to attend family events, while residents of the central area of Boston Park Street ("MA 02108'") tend to attend music events.

TF-IDF is the product of two quantities TF and IDF. The *term frequency* (*TF*) is a count of how often event j has been attended by residents in location i (it is what we called $m_{i,j}$). This count is then normalized to prevent a bias towards locations whose residents have attended a

disproportionate number of social events: $tf_{i,j} = \frac{m_{i,j}}{\sum_k m_{i,k}}$. However, to find less-attended events, we have to be more discriminating. This is the motivation behind inverse document frequency. *IDF* aims to boost events that are less frequent. If r is the number of locations, and r_j is the number of locations from which event j is attended, then *IDF* is computed as follows: $idf_j = log(\frac{r}{r_j})$. If attendees at event j come from every location, then $r = r_j$ and its *IDF* is $log(1)$ which is 0. The problem is that, in our case, *IDF* is likely to be 0 - we always find at least one resident in every location who has attended event j. That is why we modify the measure in a way similar to what Ahren *et al.*'s did [1]. We define the inverse frequency to be the inverse of the number of times event j has been attended: $idf'_j = log \frac{\sum_p \sum_q m_{p,q}}{\sum_q m_{q,j}}$. The more popular event j, the lower idf'_j.

Finally, *TF-IDF* is the prediction that users who live in location i will attend event j ($\widehat{score}_{i,j}$):

$$\widehat{score}_{i,j} = tf_{i,j} \times idf'_j \qquad (4)$$

The idea is that a location has high $\widehat{score}_{i,j}$ for the events attended more often by its residents (high $tf_{i,j}$), discounting for those events that are attended by residents in (almost) every location since they are useless as discriminators (events that have low idf'_j).

Another widely adopted approach when performing collaborative filtering is the k-Nearest Neighbor algorithm [9]. The algorithm is traditionally decomposed into the user- and item-based approaches, depending on whether neighbors are being computed for the system users or items. Our data, however, does not contain individual users; instead, it aggregates the event attendances of many users who live in the same location into a single profile. The "items" in our context are events. We thus refer to the user-based approach as k-Nearest Locations and the item-based alternative as k-Nearest Events.

Both flavors of the *k-NN* algorithm predict the extent that people living in location i will attend event j ($\widehat{score}_{i,j}$). They do so by first computing a set of similar *neighbors*. We compute the similarity between locations and events on a normalized location-event matrix N, whose elements are $n_{i,j} = \frac{m_{i,j}}{\sum_j m_{i,j}}$. The normalization ensures that all matrix values will now be in the range $[0, 1]$ and can be interpreted as the *proportion* of the number of times people from location i attend event j; we found that this process produces more reliable neighborhoods and improves the results. From the variety of available similarity measures, we selected and used the weighted cosine similarity [2].

(5) The k-Nearest Locations works by finding the top-k most similar *locations* to the one where the user lives. The similarity ($sim(i,k)$) between location i and location k is computed as:

$$sim(i,k) = \frac{\sum_e (n_{i,e} \cdot n_{k,e})}{\sqrt{\sum_e (n_{i,e}^2)}\sqrt{\sum_e (n_{k,e}^2)}} \frac{2N_{i\cup k}}{N_i + N_k}$$

The similarity shared between the two locations i and k is weighted according to the number of events N_i that users living in location i have attended (i.e., $n_{i,j}$ is non-zero), and to the number of events $N_{i\cup k}$ that users living in i or users living in k have attended. The prediction $\widehat{score}_{i,j}$ given to an event j for a user who lives in location i is computed as the similarity-weighted average of the similar locations' values:

$$\widehat{score}_{i,j} = \frac{\sum_k n_{k,j} \times sim(i,k)}{\sum_k sim(i,k)} \qquad (5)$$

(6) The k-Nearest Events. The second approach, instead, works by finding the top-k most similar *events* to the one that we are trying to predict. The prediction $\widehat{score}_{i,j}$ given to an event j for a user who lives in location i is computed as the similarity-weighted average of the similar events' values:

$$\widehat{score}_{i,j} = \frac{\sum_q n_{i,q} \times sim(j,q)}{\sum_q sim(j,q)} \qquad (6)$$

where the similarity between events j and q ($sim(j,q)$) is, as above, the weighted cosine similarity.

IV. EVALUATION

The goal of our analysis is to ascertain the extent to which one is able to recommend social events upon location information derived from mobile phone data. To this end, we need to consider a situation in which one generates for each user an ordered list of social events and then measures the quality of the newly produced list.

A. Metric for Recommendation Quality

Traditionally, collaborative filtering methods are evaluated based on the *error* between predicted and actual ratings that users input for items. Error-based metrics are not appropriate for our scenario: we are not after what the *predicted* value is for an event; instead, we are interested in whether the ranking that is produced with the predictions is useful to the user. Furthermore, in our case, a person goes to a number of social events, and only a small subset of those events are captured by our dataset. Given this sampling bias, we cannot use traditional metrics such as precision or recall. Consequently, we resort to an alternative metric called percentile-ranking [10]. The percentile-ranking $rank_{u,j}$ of event j in the list recommended to user u ranges from 0 to 1: it is 0, if event j is first in u's list; it is 1, if the event is last. Percentile-ranks have the advantage over absolute ranks

of being independent of the number of social events. Our quality measure is then the total average percentile-ranking:

$$\overline{rank} = \frac{\sum_{u,j} gone_{u,j} \cdot rank_{u,j}}{\sum_{u,j} gone_{u,j}} \quad (7)$$

where $gone_{u,j}$ is a flag that reflects whether event j was attended by u: it is 0, if j was not attended; otherwise, it is 1.

The lower $\overline{rank}$ for a list, the better the list's quality. For random predictions, the expected value for $rank_{u,j}$ is 0.5 (averaging infinite placements of event j in u's list returns the middle position of the list). Therefore, $\overline{rank} < 0.5$ indicates an algorithm better than random. We will report the results for the average percentile rankings and the corresponding confidence intervals at a confidence level of 95%. As we shall see, the intervals are very small and stay within a rounding precision of two decimal figures.

B. Area of Residence

To reproduce the cold-start situation (prediction based only on area of residence), we evaluate our six prediction techniques by using a modified version of leave-one-out cross validation. From our data, we take out one user (and the events he attended), leave all the remaining users, and predict a list of social events for the ("outed") user. We then compare how similar the two lists are: the recommended list and the list of events the user actually attended. We do so by measuring the percentile ranking as per formula (7). We repeat this process for all users (each user will be in turn removed). At the end, we compute the mean of the percentile rankings and obtain a single average percentile ranking for each technique.

Figure 1(a) shows the measured $\overline{rank}$ for the six techniques in Section III. The worst performing strategy among the six is to preferentially recommend events that are geographically close. Still, including geographic distance leads to $\overline{rank} = 0.44$ (with confidence interval [0.437,0.440]), which is lower than $\overline{rank} = 0.5$ that would be achieved by a random predictor. We achieve $\overline{rank} = 0.38$ ([0.376,0.379]) if we recommend events that are popular in general or that are attended by similar users ("k-Nearest Events") or by people living in similar areas of residence ("k-Nearest Locations"). For the two *kNN* algorithms, the results do not appreciably change as K varies. They are just slightly better if $k = 30$ for "K-Nearest Locations", and if $k = 10$ for "K-Nearest Events". The results improve if we recommend events that are popular in specific areas of residence (using techniques (3) and (4) in Figure 1(a)), which confirms our intuition that like-minded people (individuals of similar taste) tend to cluster geographically - that is, people do not "sort" themselves across areas in a random way [5]. In particular, the technique (3) which recommends events popular among residents in an area achieves $\overline{rank} = 0.33$

([0.327,0.330]) , which is a 34% gain over the random baseline.

We have tested how each technique would perform on input of only the area of residence (situation of cold-start). We now study how the six techniques perform compared to each other, and which pieces of information are valuable to make cold-start predictions. We just learned that, on average, recommending events popular among residents of the area is more effective than recommending geographically close events. This result holds, however, *in aggregate*. We now disentangle the previous results to see which technique of the six works best for 1) which event, and 2) which zipcode:

1. Which Technique for Which Event. To test how each technique performs for people who attended event j, we modify the definition of average percentile-ranking in (7) as follows:

$$\overline{rank}_j = \frac{\sum_u gone_{u,j} \cdot rank_{u,j}}{\sum_{u,j} gone_{u,j}}$$

We consider only people who have attended j. The lower $\overline{rank}_j$ for a technique, the easier to predict event j for the technique.

From our previous results, one would imagine that the best techniques are those based on: popularity (which would predict popular events); popularity in area (which would predict events popular among residents of the area); and *TF-IDF* (which would predict events popular among residents of the area, offsetting their general popularity). Figure 1(b) shows the average percentile-ranking for different events. *TF-IDF* works best for all events but the two most popular ones: "Shakespeare on the Boston Common" is best predicted by overall popularity ($\overline{rank}_j = 0.13$ with confidence interval [0.128,0.135]), and "Red Sox" by popularity in specific areas ($\overline{rank}_j = 0.28$ with confidence interval [0.284,0.288]). As one expects, to predict social events of moderate attendance, it is necessary to consider events that are popular in specific area (as technique (3) does), discounting general popularity at times (as technique (4) does). By definition, the two techniques (3) and (4) are correlated: they both recommend events popular among residents of the area, but they do so in different ways. To find out the extent to which they are correlated, we compute the Pearson correlation coefficient between their results and find it is $\rho = 0.66$, which indicates *strong* correlation: when one works well, so does the other. Instead, the correlation between the results for technique (4) and those for technique (1) (which recommends generally popular events) is weak ($\rho = 0.21$).

2. Which Technique for Which Geographic Area. To test how each technique performs for people who live in area i, we once again modify the definition of average percentile-
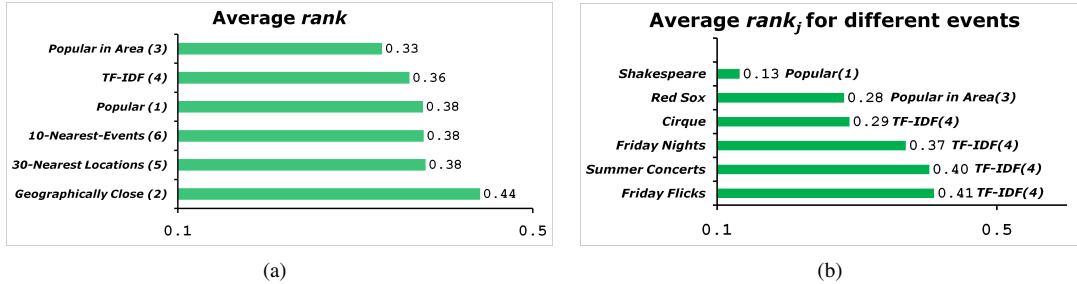
Figure 1. Performance results. The total average percentile rankings: (a) for the six recommendation techniques (numbered as per Section III); (b) for the technique (shown next to the rank value) that works best for each single event.
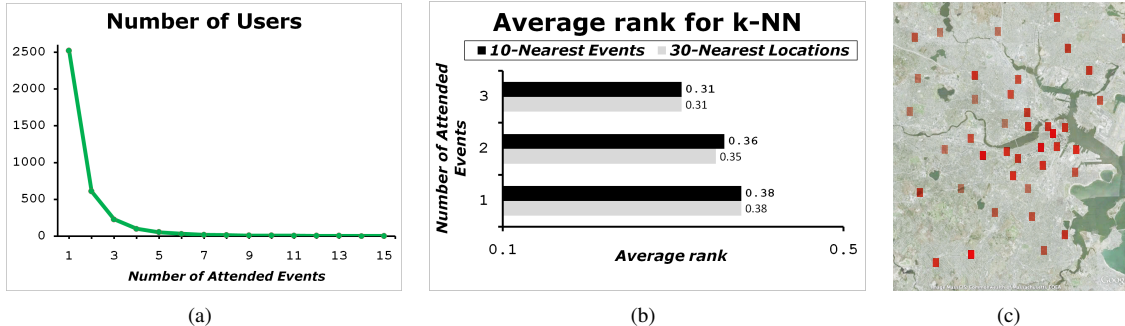

Figure 2. For people who attended a certain number of social events: (a) the number of those people (the majority has attended only one event); and (b) their average percentile-ranking for the two *K-NN* algorithms, which make predictions based not only on area of residence but also on previously attended events. The level of predictability of each location in Greater Boston from low (light red) to high (dark red).

ranking in (7) as follows:

$$\overline{rank}_i = \frac{\sum_{u,j} gone_{u,j} \cdot rank_{u,j}}{\sum_{u,j} gone_{u,j}}, \forall u \text{ living in zipcode } i$$

We now consider only people who live in area i. The lower $\overline{rank}_i$ for a technique, the easier to predict events attended by residents in area i.

By considering the best predicting technique in each area, the least predictable areas are generally suburban areas (e.g., Arlington, Charlestown, Sommerville, and Ashmont). In those cases, the best prediction is based on the most popular events. The most predictable areas are often centrally located (e.g., Boston Common, Hancock Tower). The results for an area do not depend on the number of its residents. Figure 2(c) shows a square in each location, and the intensity of the square is proportional to the extent a location is predictable. Across locations, the highest $\overline{rank}_i$ is registered for recommendations of events that are popular in one's area of residence, and the worst predictions are again those based on recommending events that are geographically close.

C. Previously Attended Events

We have evaluated the six techniques in a situation of cold-start. We have assumed that we know nothing about the user under prediction. However, in general, we would know the past events the user have attended and, based on that, we would make our predictions. To reproduce this situation, we evaluate the two *kNN* algorithms by using the original leave-one-out cross validation. From our data, we take out one single rating (i.e., one event attendance, not the whole user's rating profile), leave all the remaining ratings, and predict and rank the ("outed") rating. We then measure the percentile rankings for our prediction (we call this average rank $rank^*$). So we now make predictions upon not only the area of residence but also upon previously attended events.

The number of people who attend multiple events shows a long tail (Figure 2(a)): most people attend one event, while few people attend more than three events. As such, we obtain statistically significant results only for individuals who attend three events at most. Figure 2(b) shows that increasing the number of events that are known to have been attended by one person improves the quality of recommendations for that person. On average, by knowing only two additional events for every individual, one increases the percentile ranking by 18%. This suggests that having richer user profiles would lead to improved results. However, we also compared our approach to user-based and event-based kNN on the binary user-event (2915 x 53) matrix. We found that, when users exit the cold-start stage, a better average rank can be obtained by relinquishing the location-event dataset in favor of the user-event one: by the time a user has attended 3 events, the $rank^*$ of event-based kNN ($k = 30$) was as low as 0.08.

D. Design Consequences from Results

By analyzing the absolute values of percentile ranking, we can say little about how well each prediction technique works. However, by comparing the percentile rankings with

each other, we can say which technique performs comparatively better than the other.

In a situation of cold start (user preferences are unknown), recommending geographically close events produces the least effective recommendations, while the most effective recommendations are produced by recommending social events popular among residents of a specific area (techniques (3) and (4)). This has been assumed to be the case in previous work (e.g., by Ahern *et al.* [1], by Rattenbury *et al.* [14], by Bellotti *et al.* [4]), and we now have a quantitative backing for such an assumption with statistically significant results for a large metropolitan area. As one expects, the most popular events are highly predictable (the average rank is as low as 0.13).

If we move away from the cold start situation and consider users who have attended three events, the average rank for the user-based kNN is down to 0.08, which is comparable to percentile rankings found for recommender systems on the Web [10].

Interestingly, there are geographic areas that are more predictable than others, and this does not depend on the number of residents we consider in each area. We are trying to obtain sociodemographic data for Greater Boston to test whether sociodemographic factors such as income and inequality would explain those differences. If that would be the case, to produce effective recommendations, one would then need to complement real-time mobile data with historical sociodemographic data.

V. CONCLUSION

Upon mobile phone data, we have been able to infer which events had been attended and have considered the attendance as implicit user feedback. In studying different ways of recommending events, we found that recommending nearby events is ill-suited to effectively recommend places of interest in a city. By contrast, recommending events that are popular among residents of the area is more beneficial.

To infer attendance at social events, one needs large sets of data of location estimations. Often such sets of data are not made available to the research community, mainly for privacy concerns. Such fears are not misplaced, but they gloss over the benefits of sharing data. That is why our research agenda has been focusing on situations in which people benefit from making part of their private, aggregate data available. This paper put forward the idea that, by sharing attendance at social events, people are able to receive quality recommendations of future events.

We are currently working on: 1) designing a model for temporal tracking of events (similar to proposals made for the Web [11]); and 2) designing new privacy-preserving techniques that obfuscate user location and run directly on mobile phones [12].

REFERENCES

[1] S. Ahern, M. Naaman, R. Nair, and J. H. Yang. World Explorer: Visualizing Aggregate Data From Unstructured Text in Geo-Referenced Collections. In *ACM JCDL*, 2007.

[2] X. Amatriain, N. Lathia, J. Pujol, H. Kwak, and N. Oliver. The Wisdom of the Few: A Collaborative Filtering Approach Based on Expert Opinions from the Web. In *ACM SIGIR*, 2009.

[3] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[4] V. Bellotti, B. Begole, E. H. Chi, N. Ducheneaut, J. Fang, E. Isaacs, T. King, M. W. Newman, K. Partridge, B. Price, P. Rasmussen, M. Roberts, D. J. Schiano, and A. Walendowski. Activity-based serendipitous recommendations with the Magitti mobile leisure guide. In *ACM CHI*, 2008.

[5] B. Bishop. *The Big Sort: Why the Clustering of Like-Minded American is Tearing Us Apart* . Mariner Books, 2009.

[6] F. Calabrese, F. Pereira, G. D. Lorenzo, L. Liu, and C. Ratti. Recommending Social Events from Mobile Phone Location Data. In *Pervasive*, 2010.

[7] N. Eagle and A. Pentland. Social Serendipity: Mobilizing Social Software. *IEEE Pervasive Computing*, April 2005.

[8] J. Froehlich, M. Y. Chen, S. Consolvo, B. Harrison, and J. Landay. MyExperience: A System for In Situ Tracing and Capturing of User Feedback on Mobile Phones. In *ACM MobiSys*, 2007.

[9] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An Algorithmic Framework for Performing Collaborative Filtering. In *ACM SIGIR*, 1999.

[10] Y. Hu, Y. Koren, and C. Volinsky. Collaborative Filtering For Implicit Feedback Datasets. In *IEEE ICDM*, 2008.

[11] C. X. Lin, B. Zhao, Q. Mei, and J. Han. PET: A Statistical Model for Popular Events Tracking in Social Communities. In *SIGKDD (to appear)*, 2010.

[12] D. Quercia, I. Leontiadis, L. McNamara, C. Mascolo, and J. Crowcroft. SpotME If You Can: Randomized Responses for Location Obfuscation on Mobile Phones. In *Technical Report of The University of Cambridge*, 2010.

[13] A. Rashid, I. Albert, D. Cosley, S. Lam, S. McNee, J. Konstan, and J. Rieldl. Getting to Know You: Learning New User Preferences in Recommender Systems. In *ACM IUI*, 2002.

[14] T. Rattenbury, N. Good, and M. Naaman. Towards Automatic Extraction of Event and Place Semantics from Flickr Tags. In *ACM SIGIR*, 2007.