Semantic Place Prediction using Mobile Data

Abson Sae-Tang
École Polytechnique
Fédérale de Lausanne
(EPFL), Switzerland
abson.sae-tang@epfl.ch

Michele Catasta
École Polytechnique
Fédérale de Lausanne
(EPFL), Switzerland
michele.catasta@epfl.ch

Luke K. McDowell
Dept. of Computer Science
U.S. Naval Academy
Annapolis, MD
lmcdowel@usna.edu

Karl Aberer
École Polytechnique
Fédérale de Lausanne
(EPFL), Switzerland
karl.aberer@epfl.ch

ABSTRACT

GPS-enabled mobile devices are giving birth to a plethora of interesting applications, all based on localization capabilities. Check-ins represent a widespread trend in (mobile) social networks, but they don't exploit the full potential of mobile localization. For instance, while check-ins can provide a powerful signal of how popular a certain Point Of Interest (POI) is, they do not convey the semantic meaning of the POI. In this paper, we show how to predict (from 10 different classes) the semantic meaning of a place. Our results are obtained by careful feature extraction over a variety of data types collected by the mobile devices. With these features available, we show that a simple supervised learning algorithm can achieve reasonable accuracy on a real-life dataset provided by Nokia.

1. INTRODUCTION

This work was performed in the context of the Mobile Data Challenge launched by Nokia in early 2012. Three tasks related to mobile data mining were proposed to researchers. This paper involves the first task, which was to predict the semantic meaning of places. In particular, the task is to assign one of ten labels such as workplace, bus stop, friend's home, etc. to each place found in the data. To the best of our knowledge, although there exist many localization-based applications/services for mobile phones, such as "check-ins," none of them truly exploit the "meaning" of places. Thus, inferring the semantic meaning of places may represent an opportunity to enrich some mobile applications and make them more useful.

In the challenge data, each "place" is visited one or more times, and at each visit a variety of information such as day, time, discovered networks, applications used, and calls sent/received is collected. A training set is provided where some (but not all) of the true labels for the places are given. The final task is to predict the labels for each place in a separate test set. The test set again consists of a sequence of visits, but for a disjoint set of users.

For this problem, we chose to focus on the identification and construction of useful features that could be used for prediction, rather than on developing new prediction algorithms. Thus, we make use of standard (and simple) machine learning techniques as provided in the popular toolkit Weka [9]. In particular, we used regularized logistic regression and several simple feature selection techniques. While potentially accuracy can be improved by using more sophisticated algorithms, using simple techniques have some advantages. First, simple techniques are scalable, which could be important for mobile datasets as such datasets grow in numbers of users and time. In addition, computationally efficient algorithms are more suitable for possible deployment directly on mobile phones, instead of in a back-end server. Furthermore, simple methods tend to capture only the most relevant aspects of the underlying phenomena, leaving aside the less meaningful features, thus usually mitigating overfitting– a significant advantage for our task, given the small amount of labeled training data. Finally, as shown by Hand [2], using complex models may bring only marginal benefit at the expense of significant additional implementation and computation time.

The rest of this paper is organized as follow. The next section describes the goal pursued in this work and provides more detail about the challenge task and about what types of data were available. Section 3 describes the set of feature candidates we extracted and explains why we expected these features to be relevant for our task. In Section 4, we describe how we approached the challenge and justify the implementation choices. The performance of our method and the most interesting findings are highlighted in Section 5. Some related works are presented in Section 6. Finally, Section 7 summarizes our work and discusses potential improvements and future work.

2. TASK DESCRIPTION

This work addresses the *semantic place prediction* challenge proposed by Nokia. More specifically, the task was to lever-

1. Home
2. Home of a friend
3. My workplace/school
4. Location related to transportation
5. Workplace/school of a friend
6. Place for outdoor sports
7. Place for indoor sorts
8. Restaurant or bar
9. Shop or shopping center
10. Holiday resort

Table 1: Place labels

age mobile phone data to infer the **meaning of places** which were visited by smart phone owners. The semantic meaning of a place was represented by one of ten possible types as listed in Table 1. In machine learning terminology, because the prediction targets (classes) were known, our task results in a 10-way classification or labeling problem. Each place to be classified was represented by one or more visits. A partially labeled training set was provided, and the final objective was to classify places in a test set that was constructed from a separate set of users.

Unlike most prior work in this area [3][4], for privacy reasons, we did not have access to GPS information. Instead we were provided with **symbolic place IDs** which have been inferred partly based on raw GPS data. Although no localization information was available, we had the *visit history* of places, which means we knew when a user arrived at a given place and when the user left. Notice that because GPS data was not given, and because place identifiers were made user specific, it was not possible to determine if two users visited the same place.

In addition to the information about visit sequence and timing, we also had access to a broad range of contextual data from the users' smart phone. For instance, data was provided on the `wlan` networks detected, the `calls` made by the user, the mobile `applications` used, etc. In general, the specific values for these attributes (such as the names of networks) were also anonymized in a user-specific way.

3. FEATURE EXTRACTION

Guided by common sense and preliminary data analysis, we extracted a large number of features that we expected would be relevant for the place label prediction. This section describes those features, while Section 4 describes how a smaller set of features was automatically selected for actual use by the classifier.

We considered two general categories of features. The first one, based only on the *visit sequence* dataset could be used on its own, without the need to reference with (or *join*) any other data. For instance, time of day is a simple feature in this category. The second group of features, which we will refer to as *mobile* data, was derived from phone logs that needed to be joined with the visit sequence data in order to be useful. For instance, the number of calls received was one such feature in this category, because the call log needed to be joined with the visit sequence in order to connect calls to their associated visit.

In the training set, only about 66% of all visits were labeled as having `trusted_start` and `trusted_end` times. However, in order to keep the training set as large as possible, we treated all start and end times as trusted.

3.1 Visit sequences

Participant visits were recorded in two datasets: `visit_sequence_10min` and `visit_sequence_20min`. These datasets contain all visits of at least 10 (respectively, 20) minutes. The second dataset is thus a proper subset of the first, except that it also contained an additional flag `trusted_transition`. Because we did not believe that this flag would be helpful for prediction, we chose to utilize only the first, larger dataset `visit_sequence_10min`.

Clearly some key characteristics relevant to the semantic meaning of a place are the temporal properties of visits to that place. The visit sequence data provided information about when a user arrived at a place (`start_time`) and when the user left (`end_time`). Early examination of the data convinced us that the *visit duration* was a good candidate feature. For instance, we observed that most people stayed at their `workplace` for at least four hours, and even longer for their `home`.

Likewise, visits at certain periods of the day should also be good indicators for the meaning of a place. For example, most people have lunch around noon, so visits between 11:45 and 13:30 are likely to be to `restaurant` places. Likewise, the day of the week of visits should also supply information on the type of places visited. For instance, in Switzerland (where the data collection campaign took place) opening times are strictly regulated by laws, so it is very unlikely that a user visits a `shop` place on Sunday.

To capture these temporal trends, we first divided the week into three type of days: *Weekday, Saturday* and *Sunday*. Then, as shown in Figure 1, we built a different time discretization for each type of day. The bins represent the starting time of a visit. For instance, visits starting on a *Weekday* can fall into 9 different bins, and a visit starting between 7 a.m. and 10 a.m. will be assigned to bin number 3.

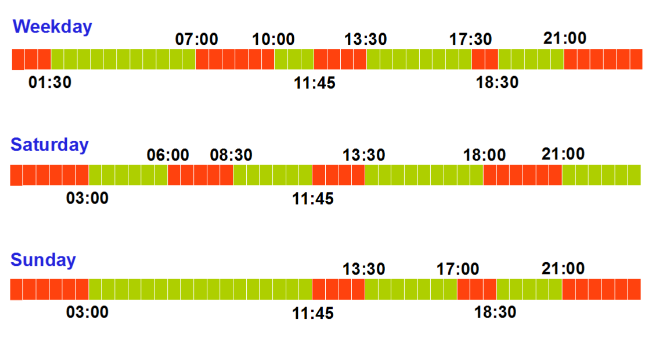

Figure 1: Day discretization

Each place can then be represented by a number of features that are computed by counting with these bins. For instance, for a certain place p, counting the number of visits that are assigned to each of the 9 *Weekday* bins yields 9 features that describe that place. We also computed features over coarser granularities, e.g. the number of visits for each type of day (weekday, Saturday, or Sunday) and the total number of visits (i.e., the “global” number of visits).

These features, combined with a few others based on duration and counting of distinct occurrences, yielded the following set of candidates:

- nb. of visits global
- nb. of visits on weekdays
- nb. of visits on Saturday
- nb. of visits on Sunday
- nb. of visits in weekday bins [1..9]
- nb. of visits in Saturday bins [1..8]
- nb. of visits in Sunday bins [1..7]
- duration of visits global
- duration of visits on weekdays
- duration of visits on Saturday
- duration of visits on Sunday
- nb. of distinct bins visited on weekdays
- nb. of distinct bins visited on Saturday
- nb. of distinct bins visited on Sunday

Note that we are not classifying individual visits, but rather places that are each described by a set of visits to that place. Thus, each of the features listed above must be an aggregation over a set of visits that are all associated with one particular place. To account for the fact that some places may be visited more frequently than others, we computed two versions of each feature. The first version was the corresponding feature value for a visit, summed over all visits (e.g., the total number of visits, or the total time duration of some visits). We refer to these as “number” or “sum”-type features. The second version was the same feature value, but normalized (averaged) over the appropriate number of visits. We refer to these as “mean”-type features. Finally, we also computed the standard deviation of each feature value and provided this as a candidate feature (though our results later show that none of them were selected by the feature selection technique). The computation of sum, mean, and standard deviations was also applied to the other features described in the following sections.

3.2 Mobile data

This section describes features that were constructed by joining together the visit sequence data with some additional data collected by the user’s phone.

3.2.1 Wlan

We noticed that for some places, like `work` or `home`, the *number of distinct wlans* detected is often quite high. To create a feature based on these detections, while filtering out some noise, we used the following strategy. First, we observe that wlan access points are identified by their `ssid` and rarely change, so each place should be roughly characterized by a set of ssids. To assess (and then count) which ssids are relevant to a specific place, we retrieved a list of ssids seen at that place during all the visits. Then we filtered out those ssids which were observed less than n times, considering them as being noisy detections (we considered $n = 2, 3$, or 4). Finally, we counted, for a particular place, the number of remaining distinct ssids.

Note that because ssids were anonymized by user, we do not know when the same wlan network is seen by different users. Therefore it is not possible to leverage the wlan data to extend the ground truth by sharing labels among users.

3.2.2 GSM and Bluetooth

As with the wlan networks, we extracted (by place and by user) the *number of distinct bluetooth* devices (identified by their `mac_address`) detected over all visits. Such features could help to identify places where there are many other people around, e.g., at `restaurants and bars`.

Likewise, for GSM data we counted the *number of antennas* (or `cell_ids`) seen at a given place. We thought that this feature could help to identify areas which are sparsely covered by GSM, such as countrysides or nature parks where people might participate in `outdoor sports` or spend their `holiday`.

3.2.3 Accelerometer

The field `avdelt` from the accelerometer dataset averages, over a sampling period of about 50 seconds, the *difference in the acceleration* between two consecutive measures for each of the three axes x, y, and z. To roughly measure behavioral differences, we created a feature for each place that counted how many time over all the visits the `avdelt` was above a specific threshold. Initially, we used a user-specific threshold that was computed by identifying the 90th percentile of a user’s `avdelt` values. However, further analysis showed that this value was very similar across most users (especially compared to the maximum possible value), so we ultimately decided to use a single threshold across all users.

3.2.4 System

Users are much more likely to charge their cell phone at `home` or at `work` then at a different location. To detect such places, we computed two *charging frequency* features. The first one calculates the ratio between the number of charging actions (when the `charging` status goes from 0 to 1) and the number of visits. The second feature normalizes the number of charging events for a given place by the total number of charging events for all places.

Users may also vary the current phone profile based on their activity and/or place. For instance, they may use silent mode at a `restaurant` but tend to leave the phone in normal mode at `home`. However, the specific preferred phone `profile` (or ring) for a given place will depend upon the user’s habits. Thus, to create a feature that was more robust to user variations, we computed the number of *profile switches* per visit, normalized by the visit duration. Similarly, phone inactivity may also be correlated with place. To measure such effects, we computed two features: the *longest inactivity period* and *the sum of the inactive periods.*

3.2.5 Application

In order to identify possible relationships between places and the use of mobile applications, we first identified the most popular applications over all users, by counting for each of them the number of `start_events`. Then we clustered the frequently used applications into thematic classes that seemed relevant for the prediction task: *clock, phone book, calendar, camera, map, email, web, multimedia*, and *leisure time* (which includes email, web and multimedia). Finally we computed for each place how often each type of application was used (again, normalized by visit duration).

We ignored some frequent "applications" such as *Standby mode* or *Application grid* because they did not seem likely to be correlated with the semantic label of a place. We also ignored applications related to calls and messaging, because they are likely to be redundant with the features based on call log data that are described below.

3.2.6 Call log

The call log dataset records incoming and outgoing communications for both voice calls and short messages (SMS). Furthermore, it provides information on call durations and missed calls. We considered independently the outgoing and incoming communications, although knowing how they relate to each other (e.g., their relative proportion) could also be informative.

We extracted the following list of features (here communication stands for calls + SMS): *number of outgoing calls*, *number of incoming calls*, *number of missed calls*, *number of SMS sent*, *number of SMS received*, *outgoing call duration*, *incoming call duration*, *number of outgoing communications*, *number of incoming communications*, *in-out call ratio*, *in-out SMS ratio*, *in-out call duration ratio*, *in-out communication ratio*. All of these feature are normalized by the visit duration.

4. METHODS

As described in Section 1, we decided to focus on task-specific feature construction coupled with standard machine learning techniques for feature selection and classification. In particular, we used Weka[1], a very common machine learning library. Weka provided also the tools necessary to do cross-validation and to combine the feature selection with the classification.

Given that the sets of users from the training vs. test data were disjoint, we chose to use a single classifier that would learn global patterns across all the users. For the classifier, we selected *multinomial logistic regression* due to its simplicity, typically strong accuracy, and ability to tolerate correlated features. Because we had a large number of features and a relatively small labeled training set, overfitting was a potential concern–we addressed it in two steps. First, to regularize the model, we assumed a Gaussian prior on the feature weights learned by the model; the variance of these priors was controlled by setting the "Ridge" parameter. Second, we performed feature selection to reduce the number of features actually used by the model. In particular, we evaluated both *filter* and *wrapper* methods for feature selection [6]. The filter method we chose (`CfsSubsetEval`) looks at each feature individually and then selects the one that has a high correlation with the class to predict, but a low correlation with the features that have already been selected. The wrapper method simply uses our main classifier to evaluate the resultant accuracy of each subset of features by cross-validation. We used the `LinearForwardSelection` algorithm to search over the feature space. This search method is relatively fast, and produces a compact final feature subset.

[1] `http://www.cs.waikato.ac.nz/ml/weka/`

In the training data, only 336 of the 6350 visited places (5.3%) were labeled, thus forming the ground truth. Given this sparsely-labeled scenario, semi-supervised learning (SSL) could be a viable option to improve the prediction accuracy. In particular, two SSL techniques that we considered for our task were *self-learning* and *co-training* [10]. The former technique seeks to iteratively estimate labels for the unlabeled training instances, identify the instances with the most confident predicted labels, then retrain the classifier with the labeled data and the most confident inferred labels. The co-training technique involves two classifiers which are trained on disjoint features sets and are then used to provide additional (high confidence) labeled examples to each other, gradually extending the size of the usable training set. Although there were also multiple possible ways to divide our data features into two sets, SSL is known to be challenging to use, and to not always improve performance. Section 5 briefly considers some use of SSL.

5. RESULTS

We evaluated the overall accuracy using 10-fold cross-validation on the provided training data. Surprisingly, the set of features selected by the filter method differed only slightly from the ones selected by the wrapper. Both the techniques produced on average the same overall accuracy, therefore we present detailed results only for the wrapper method.

5.1 Selected Features

Figure 2 shows the 16 features that were selected by the wrapper method. Notice that most of them (12) come from the visit sequence. This outcome seems to indicate that there exists a strong and global relationship shared by many users between the temporal dimension and some of the types of places visited, which is perfectly aligned with our first intuition. On the other hand, it appears that we did not manage to identify many good features from the "mobile" data. We suspect that this is due in large part to the sparsity of these features: at least for this training data, for many places the values of the mobile features were predominantly zero, e.g., because no communication events took place or no applications were used at that place. This data sparsity may be partly explained by the fact that at least some users participating in the data collection campaign were simultaneously using two cell phones (their regular phone plus the one provided by Nokia), and thus not all relevant data was collected. It is also possible that the mobile features that we created were not well-aligned with the prediction task. Alternatively, it may be that our mobile features *could* be strongly predictive of a place, but only when a user-specific classifier was used, instead of the global classifier necessary for this task.

A closer look at the extracted features reveals that 62.5% of

- nb. of weekday visits starting in [00:30 - 03:00]
- **nb. of weekday visits starting in [07:00 - 10:00]**
- **nb. of weekday visits starting in [18:30 - 21:00]**
- **nb. of Saturday visits starting in [18:00 - 21:00]**
- nb. of Sunday visits starting in [00:00 - 06:00]
- **nb. of visits on weekdays**
- **nb. of visits on Saturday**
- **nb. of visits on Sunday**
- nb. of bins visited on Sunday
- **global visit durations sum**
- global visit durations mean
- **weekday visit durations sum**
- nb. of activity peaks (accelerometer)
- **nb. of bluetooth mean**
- nb. charging events / nb. visits
- nb. charging events / tot. charging event

Figure 2: Features selected by the wrapper feature selection method. Features shown in blue/bold were also selected by the filter method.

them (those highlighted in Figure 2) were found by both the wrapper and the filter methods, which suggests that they are strongly discriminative. This high overlap is possibly linked to the search method used (which was the same), but also to the classifier employed. Since logistic regression is a linear method, it will tend to favor features which can linearly explain the labels.

5.2 Classifier Accuracy

Using cross-validation on the training set, the logistic regression classifier achieved an average accuracy of 67.6% (227 correct out of 336 places). In contrast, simply guessing the dominant class (workplace/school) would achieve an accuracy of only about 30%. We also learned classifiers for two different subset of the features in Figure 2: one that contained only *visit sequence* features, the other one with only *mobile* data (as explained in Section 3). We obtained, respectively, an average accuracy of 64.2% and 58%.

Thus, the combination of our features and classifier succeeded in obtaining informative predictions about the semantic meaning of many places. On the other hand, this accuracy fell short of our initial goals, as discussed later. We also note that the small size of the labeled training set means that this overall accuracy number may not be very reflective of the performance on a larger sample.

Table 2 shows results broken down by the true class label of each place. Clearly, the highest precision and recall are achieved on `home` and `my workplace` (with precision of 0.91 and 0.756, respectively). Strong performance on these two classes is not very surprising, since it is relatively easy to design features that are highly predictive of these locations–users make many visits to them, and we had the most training data for these classes (see Figure 3). In

Label	Place	Precision	Recall
1	Home	0.91	0.845
2	Home of a friend	0.537	0.63
3	My workplace/school	0.756	0.882
4	Location related to transp.	0.375	0.391
5	Workplace/school of a friend	0	0
6	Place for outdoor sports	0.4	0.48
7	Place for indoor sports	0.333	0.286
8	Restaurant or bar	0.333	0.091
9	Shop or shopping center	0.688	0.647
10	Holiday resort	0	0

Table 2: Classification results summary by class.

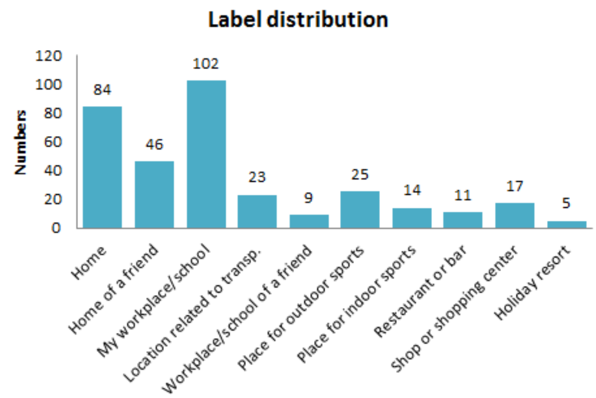

Figure 3: Number of training labels for each class.

contrast, the classifier performs very poorly on the classes for which there are few training data instances, such as `holiday` and `friend's workplace`. In most cases there is a strong correlation between the performance of the classifier and the amount of labeled examples available.

Table 3 is a confusion matrix that helps to explain the nature of the errors made by the classifier. For instance, row #2 of the table shows that 4 places with true class #2 were predicted as class #1, and 29 places with true class #2 were predicted correctly (see Table 2 for the numbering of the classes). Interestingly, most of the `friend's workplace` places were classified as `my workplace`, which is an understandable error. Similarly most of the misclassifications of `home` places went to the `friend's home`. Finally, perhaps because they were geographically close, `places related to transportation` were often confused with `workplace`

The recall for `restaurant` was surprisingly low, despite the fact that the day discretization we designed effectively provided a dedicated bin that should have been predictive of this class. This result can probably be explained by the fact that the classifier was not able to leverage enough mobile data to discriminate `restaurant` places from other places like `indoor sports` which were visited in the same time span. Indeed, Table 3 shows that such prediction errors did occur.

We investigated SSL techniques only briefly, but did consider

	↓ Classified as ↓									
	1	2	3	4	5	6	7	8	9	10
1	71	8	2	0	0	1	0	1	1	0
2	4	29	2	4	0	5	1	1	0	0
3	3	5	90	3	0	0	1	0	0	0
4	0	1	7	9	0	3	1	0	2	0
5	0	0	8	1	0	0	0	0	0	0
6	0	4	2	3	0	12	2	0	2	0
7	0	2	4	0	0	4	3	1	0	0
8	0	3	3	0	0	2	2	1	0	0
9	0	1	0	4	0	0	1	0	11	0
10	0	1	1	0	0	2	1	0	0	0

Table 3: Confusion matrix. The true label of each example is indicated in the leftmost column.

one form of self-learning. Because our classifier achieved strong accuracy only on `home` and `my workplace`, we decided to augment the ground truth only with predicted instances of those two classes, though we recognize the potential bias that such a choice could induce. The label extension process considered each class after the other, starting first with `home` followed by `my workplace`. At each iteration only the 7 most confident predictions were promoted to the status of a regular label, provided that they had a confidence score of at least 85%. Given these restrictive conditions the algorithm stopped after 3 iterations and then produced an overall accuracy of 68.2%, which was a small improvement. These gains came mainly from the `outdoor sports` class, which improved to a precision of 0.5 and a recall of 0.6. Thus, SSL improved performance but only slightly and not necessarily in a statistically significant way. We suspect that SSL may offer more gains for this task if the baseline accuracy of the classifier could be improved or if more sophisticated SSL techniques were considered [8].

For completeness, we also tried a small variant for counting the *number of visits* that fall into each bin (see Section 3.1) where the counting considers the entire visit period instead of only the start of the visit. Unfortunately this variant did not bring any improvement.

6. RELATED WORKS

There is a vast literature on algorithms and systems that are able to detect and learn the important places for a user [3][4]. Depending on the context, GPS data can lead to good detection accuracy [7], even though the signal is unreliable in closed environments. Wi-Fi, GSM and historical data can play a key role in improving the detection accuracy [5], but they have still not be used to infer the semantic meaning of a place.

To our knowledge, this is the first work that tackles a scenario where both accurate GPS data and user behavioral information are missing. Furthermore, previous works focused only on classes with coarser granularity (e.g., just "home", "work" and "other" [1]).

7. CONCLUSIONS

We were expecting to be able to predict the semantic meaning of places with an higher accuracy than what we achieved– we now discuss a few factors that could explain why we did not reach our goal. First, many of the provided data signals were more sparse than anticipated, perhaps due to the previously discussed simultaneous use of multiple phones (see Section 5). Second, because the users in the test set did not appear in the training set, we could not do user-specific predictions; we expect that capturing more user-specific behaviors could help substantially. Finally, having a fairly small set of known training labels proved to be a very challenging scenario, and our results suggest that classes with more training labels were much easier to predict (see Table 2).

Nevertheless, we did manage to identify a combination of features and classifier that substantially improved over a random classifier, and our results show that temporal properties are especially helpful for semantic place prediction. Future work should begin by evaluating the kinds of features that we have described on a dataset with many more labeled examples, to see how performance improves and to assess the extent to which our results generalize. In addition, we would like to explore the development of new mobile features to better connect place prediction with events like phone application usage and communications sent/received. Finally, given that labeled examples are often difficult to obtain, but unlabeled examples are fairly easy to collect from mobile devices, we would like to explore in more depth the use of semi-supervised learning for this domain.

8. REFERENCES

[1] R. Ahas, S. Silm, O. Järv, E. Saluveer, and M. Tiru. Using mobile positioning data to model locations meaningful to users of mobile phones. *Journal of Urban Technology*, 17(1):3–27, 2010.

[2] D. J. Hand. Classifier Technology and the Illusion of Progress, June 2006.

[3] J. Hightower, S. Consolvo, A. LaMarca, I. E. Smith, and J. Hughes. Learning and recognizing the places we go. In *Ubicomp*, volume 3660 of *Lecture Notes in Computer Science*, pages 159–176. Springer, 2005.

[4] J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello. Extracting places from traces of locations. *Mobile Computing and Communications Review*, 9(3):58–68, 2005.

[5] A. Kirmse, T. Udeshi, P. Bellver, and J. Shuma. Extracting patterns from location history. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 397–400. ACM, 2011.

[6] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artifical Intelligence*, 97(1-2):273–324, 1997.

[7] L. Liao, D. J. Patterson, D. Fox, and H. Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 171(5–6):311–331, 2007.

[8] G. Mann and A. McCallum. Simple, robust, scalable semi-supervised learning via expectation regularization. In *Proc. of ICML*, pages 593–600, 2007.

[9] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.

[10] X. Zhu and A. B. Goldberg. *Introduction to Semi-Supervised Learning*. Morgan & Claypool, 2009.