Indoor-ALPS: An Adaptive Indoor Location Prediction System

Christian Koehler, Nikola Banovic, Ian Oakley[1], Jennifer Mankoff, and Anind K. Dey
Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh PA, 15213
ckoehler,nbanovic,jmankoff,anind@andrew.cmu.edu
University of Madeira & UNIST[1]
UNIST, UNIST-gil 50,
Ulsan 689-798, Republic of Korea
ian.r.oakley@gmail.com

ABSTRACT
Location prediction enables us to use a person's mobility history to realize various applications such as efficient temperature control, opportunistic meeting support, and automated receptionists. Indoor location prediction is a challenging problem, particularly due to a high density of possible locations and short transition distances between these locations. In this paper we present Indoor-ALPS, an Adaptive Indoor Location Prediction System that uses temporal-spatial features to create individual daily models for the prediction of when a user will leave their current location (transition time) and the next location she will transition to. We tested Indoor-ALPS on the Augsburg Indoor Location Tracking Benchmark and compared our approach to the best performing temporal-spatial mobility prediction algorithm, Prediction by Partial Match (PPM). Our results show that Indoor-ALPS improves the temporal-spatial prediction accuracy over PPM for look-aheads up to 90 minutes by 6.2%, and for up to 30 minute look-aheads by 10.7%. These results demonstrate that Indoor-ALPS can be used to support a wide variety of indoor mobility prediction-based applications.

AUTHOR KEYWORDS
Location-based Services, Machine Learning

ACM CLASSIFICATION KEYWORDS
H.m. Information Systems: Miscellaneous

INTRODUCTION
The ability to capture people's location within large and complex indoor spaces (such as office buildings, university campus buildings, or hospitals) and to use this information to predict when and where they will go next is a powerful computational tool. Indoor location prediction algorithms could enable smarter automated receptionists [5] that could help visitors know when a building occupant is likely to return to her office, or arrange ad-hoc meetings whilst occupants are transitioning between locations [13].

UbiComp'14, September 13-17, 2014, Seattle, WA, USA

ACM 978-1-4503-2968-2/14/09…$15.00.
http://dx.doi.org/10.1145/2632048.2632069

However, predicting occupant indoor locations and when occupants will transition between these locations is challenging. Existing systems that monolithically approach this task, combining a prediction of where a user will go with one of when they will go there, perform poorly with prediction results that are close to the performance of a majority predictor (*i.e.*, predicting that the occupant will always stay in his most frequented location) [3,19]. This is likely because occupants' routines within the interior of a building, such as a workspace, can be very complex. They can change substantially from weekday to weekday, and encompass numerous destinations and transitions in relatively short spans of time.

To simplify this problem, most prior work has focused on predicting an occupant's next location without taking into account when these transitions will take place [12,14,15,22,23]. Solving this problem alone precludes many applications that rely on knowledge about transition timings. For example, a system that proactively heats a room in advance of an occupant's arrival [20] would not know whether the person was coming in 5 minutes or 2 hours, and could result in either significant discomfort (if heated too late), or significant wasted energy (if heated too early).

One factor that makes it difficult to accurately predict an occupant's transition time into a space is the fact that indoor routes are typically short (meters to hundreds of meters) and traversed rapidly (in seconds to minutes). This is in contrast to the problem of outdoor location prediction (*e.g.*, [9,24]), for which travel times and distances are long enough that a useful prediction algorithm only needs to predict the next significant location after the user has already departed. This allows the use of features such as the currently traversed path to make predictions, which are mostly not useful to applications of indoor prediction, due to the shorter indoor transition times. This also means that many of the analytic techniques that perform well outdoors are inappropriate, or simply do not work well, when applied to indoor scenarios.

Making exact predictions about when a person transitions is a hard problem due to the variability in length of stay at a location. How long a person stays at a given location is dependent on their temporal-spatial routine and can vary greatly from weekday to weekday and fluctuates depending on prior events that occur during a day. Considering the difficulty of this problem, even given its importance for different applications, we instead focus on an important sub-problem: predicting if an occupant will stay at their current location for a specified time frame and (if they are predicted *not* to stay) predicting the indoor location they will transition to within that time frame. We chose this problem because it is complex, including both temporal and spatial prediction aspects, and it is powerful enough to allow us to support a large class of compelling applications.

For example, with our proposed predictor, we could build an application that would allow you to more easily have *ad-hoc* meetings with co-workers [13], by knowing if they were going to be in their current location for at least the next 20 minutes. We could also improve automated receptionists like Roboceptionist [5] that can tell visitors if an office occupant will return to her office within the next ten minutes. Furthermore, we could enable smarter building control systems capable of optimizing energy consumption based on occupants' movement patterns (*e.g.*, by realizing zoned predictive temperature control) [20]. Such systems could minimize operational costs while also limiting the impact on occupant comfort [11]. Finally, with a prediction that an office worker is leaving within 30 minutes, a notification system could issue a warning that traffic is especially bad and that she should leave early to arrive at her destination on time.

To support these and other similar applications, we propose a novel temporal-spatial indoor location prediction algorithm called Indoor Adaptive Location Prediction System (*Indoor-ALPS*). Indoor-ALPS tackles the challenge of accurate temporal and spatial prediction by splitting the problem into two separate steps: 1) For a given time interval, Indoor-ALPS predicts whether an occupant will stay in her space for at least that long and 2) for a given location, Indoor-ALPS predicts the next location she will transition to. The algorithm then combines these two independent predictions. It first predicts whether or not an occupant will stay at a location for at least a given time window and then predicts her destination. Unlike previous approaches that perform these two steps simultaneously, Indoor-ALPS's decomposition of the problem allows it to learn which contextual features are the best predictors for the individual temporal and spatial problems. We argue that dividing the problem in this way will lead to improved prediction performance.

To verify this claim, we compared Indoor-ALPS to Prediction by Partial Match (PPM), a state of the art indoor location prediction algorithm [3]. We evaluated our approach using a range of time windows from 10 to 90 minutes, in an effort to show its applicability for different applications. Our analysis showed that Indoor-ALPS improved the overall prediction accuracy by 6.2% over PPM. Indoor-ALPS was particularly strong when considering temporal look-aheads of 10 to 30 minutes, when it led to a significant mean accuracy improvement of 10.7% with a maximum improvement of 12.9%. This paper presents two contributions: the first contribution is a new algorithm that addresses the temporal aspect of the indoor location prediction problem and the second contribution is a hybrid algorithm that addresses the combined problem of when will an occupant transition and to where. With this prediction algorithm, a number of compelling applications that rely on indoor mobility prediction can be supported.

RELATED WORK

Indoor location prediction most commonly focuses on the problem of predicting where a person will go next. To make such a prediction, we must define a set of *significant locations*. A location is considered significant if a person frequently spends at least ten minutes at that location [1]. Because of the short distances between significant locations (many buildings can be traversed in about five minutes), it is necessary to predict where a person will go next *before* they start moving. This prevents us from using algorithms that have been successful in outdoor settings, which make predictions based on the current path an occupant is traversing [9,24]. It also increases the difficulty of the prediction problem, because ideally we will want to predict not only *where* the occupant is going next but also *when* they will go there.

Given these challenges, prior research has addressed the problem of indoor location prediction by concentrating on three different prediction sub-tasks: predicting whether a specific location will be occupied at a specific time in the future (*e.g.*, [20]); predicting *where* an occupant will go next without considering *when* the transition will occur (*e.g.*, [12,14,17,22,23]); and predicting *where* an occupant will be at a specified time (*e.g.*, [3,19]). The majority of these algorithms focus on the problem of predicting a person's next location irrespective of time. Even though an important problem for many applications, it is also valuable to have knowledge about when a person will transition and where he will transition to. To our knowledge only two algorithms have tackled this latter problem for indoor location prediction. Following are the key algorithms that attempt to address the three prediction sub-tasks.

An algorithm solving the task of predicting if an individual *room will be occupied or not* given a certain discrete time stamp does not focus on occupant movement or the occupancy status of other rooms. One example of such a system is PreHeat [20], an occupancy prediction algorithm intended for residential use. PreHeat was studied by collecting data from five households using RFID tags on the occupants' keychains and predicting when the home (all five households) and/or individual rooms (for two of the households) would be occupied. The collected data was formatted into discrete 15-minute timeslots and the algorithm used the partially observed day (up until the current timeslot) to identify and retrieve the five most similar days (in terms of occupancy) from the history of all prior days. The occupancy of a future timeslot *t* was calculated as a majority vote over the timeslot *t* of the five retrieved days. The median prediction accuracy was above 80% for the whole-house occupancy prediction. The authors also showed that PreHeat tracked the occupancy of individual rooms fairly well with only small differences between the actual and predicted occupancy.

An algorithm that predicts users' *next location irrespective of time* typically takes their current location and location history as input to make a prediction about their immediately subsequent location. This prediction is done irrespective of the current or future time, meaning the approach does not attempt to make inferences about when the user will transition to the new location. It could be in five minutes, in one hour or even longer. Petzold *et al*. [17] analyzed a range of different algorithms for their suitability for next place prediction. They compared 7 different algorithms: Bayesian network, multi-layer perceptron, Elman net, Markov predictor, state predictor, Markov predictor with confidence counter, and state predictor with confidence counter. They compared the algorithms on the basis of criteria such as accuracy, learning and relearning speed and cost. They achieved a maximum accuracy of 79.68% using an Elman Net. Using the two algorithms with a confidence counter they were able to achieve a slightly higher accuracy of 81%, but only by withholding one-fourth of the prediction results for which these approaches had low confidence. The analysis of the other metrics also showed that the higher performance of the best algorithms (Elman Net and multilayer perceptron) came at the cost of slower learning and relearning speed and high modeling overheads.

Vuong *et al*. [23] introduced the Adaptive Confidence Estimator (ACE) for next location prediction. ACE used a confidence counter to express how confident the algorithm is with a prediction result. The authors reported an average accuracy of 88.57% after withholding about one-fourth of the prediction results due to low confidence (similar to the Petzold work). Another approach to the next location prediction problem is to use neural networks [22]. This resulted in an accuracy of 76.4% accuracy, with 92.3% accuracy on the binary prediction of whether or not the next location will be an individual's office.

The task of *location prediction given a particular time* predicts users' location given a specific time stamp and their time-stamped movement history, but does not predict when the user transitions. To address this prediction sub-task, Ryan and Brown [19] used an Association Rule Mining approach. Specifically, they were predicting the next location given a look-ahead time window. To test their approach they used an internally collected dataset as well as the Augsburg Indoor Location Tracking Benchmark. Their approach achieved an accuracy of up to 79% when predicting a user's location with a look-ahead time of one hour on their dataset. However, when applying their algorithm to the Augsburg dataset, the accuracy dropped to 56%. The authors accounted for this drop by explaining that their algorithm assumes that a sequence of visited locations always occurs at the same time and thus shifts in the time of transitions are problematic for the algorithm.

Another approach to this problem uses a text compression algorithm, Prediction by Partial Match (PPM) [3]. PPM predicts the most frequent location for a given timestamp using only a discrete 10-minute time slot. It does not use the temporal distance between the current and future time stamp nor the current location or any other additional features. It was evaluated on the University of California, San Diego Wireless Topology Discovery dataset, a publicly available dataset that tracked the indoor movements of 300 freshmen college students using the university's Wi-Fi access point network. The reported average accuracy for PPM on the UCSD dataset was 87%. Since PPM has demonstrated the highest reported accuracy on the prediction problem we are most interested in, we chose to evaluate our approach against it.

The majority of the effort in location prediction has been on addressing next place prediction. As there are a class of applications that rely on not only having a prediction of where someone will go next but also when she will transition, we focus our efforts on this combined temporal-spatial prediction problem. As described in this section, the best performing model, PPM, only uses the prediction time slot and a histogram of significant locations. We see an opportunity to improve on this work, by leveraging additional temporal and spatial information commonly available in indoor mobility datasets.

INDOOR-ALPS: A HYBRID ALGORITHM

As mentioned in the introduction, Indoor-ALPS splits the prediction problem into two independent steps, which it later combines. First, for a given time interval, Indoor-ALPS predicts whether the user will *stay for at least that long* in the current location. Second, for a given location, it predicts where the user will go next. Finally, Indoor-ALPS combines these predictions.

More specifically, to make both our spatial and temporal predictions for each day, we use the same algorithm (independently for each type of prediction). It uses a combination of two approaches that can help improve classification accuracy: *ensemble prediction* (training and combining results from several classifiers); and *incremental learning* [6] (using each newly recorded data point as part of the training dataset after the prediction for that data point is completed). The basic algorithm is as follows:

1. Take all the data in the dataset up until the current day (for which prediction is being performed), and split it into two non-overlapping data sets: optimization and training.
2. Using this data, identify the best feature-subset for each classifier used in the ensemble algorithm:

Decision Tree, 3-Nearest Neighbor, Support Vector Machine, and Gradient Boost.

3. Train each of the four classifiers, using their feature set tailored to them in Step 2.
4. Using each classifier, make predictions for the current day.
5. Calculate the ensemble prediction [11] using the results of each individual classifier.
6. Repeat from step 1 until all days in validation set are predicted.

We use a portion of the data for each user to create initial models and validate on the remaining data for that user. Initially 10 days of data are used and equally split into the optimization and training data set. After feature selection using the optimization data set, the models for each algorithm are created using the training set and predictions are performed on the validation day. After the predictions are complete for that day, that day's data is added to either the optimization or training data sets. These days of data are added in an alternating fashion, first day to the training set, next day to the optimization set, and so on, to maintain an almost-even split of the data between optimization and training.

Feature Selection

Predicting if a person stays at a location for a given amount of time and where they transition to if they do not stay is highly dependent on their temporal-spatial routine. Making a temporal or spatial prediction depends on factors such as day of week, current location, or even the arrival time at the office in the morning. Indoor-ALPS uses ten temporal-spatial features that capture these influences:

- Current location (*L*)
- Time of arrival at *L*
- Minutes passed since arriving at *L*
- Current time
- Current day of the week
- Arrival time in the building for the current day
- The number of significant locations the occupant visited previously for the current day
- Previous two significant locations
- Stay duration at the previous significant location
- Length of the transition time to the current location

All times are expressed as discrete ten-minute timeslots starting from midnight and significant locations are defined as locations in which the occupant frequently stays for at least ten minutes [1].

Since we do not know if all features are equally important for each user and situation, our algorithm uses Sequential Floating Forward Selection (SFFS) [18] to find the most relevant features from the set of ten features. SFFS is a greedy algorithm that adds one feature per iteration to the already selected feature subset. After each new feature has been selected SFFS checks whether a subset of already selected features can be removed without decreasing the performance. We used an objective function that maximizes accuracy. We allowed SFFS to create a feature set consisting of between one and ten features based on this objective function. Our algorithm applied SFFS independently for four different machine-learning algorithms: Decision Tree, 3-Nearest Neighbor, Support Vector Machine, and Gradient Boost.

Our algorithm re-evaluates, after each predicted day what is the best feature subset for each algorithm. By applying this incremental learning approach, the algorithm has the opportunity to react to changes in users' temporal-spatial routines.

The location data used for the prediction is formatted into discrete 10-minute time slots, both to make it easier to compare our algorithm against previous techniques [3] and to keep in line with the definition for a significant location [1]. For each stay at a location we interpolated data points that represent the current stay duration at a location. For example, let us assume that the user arrived at *Location A* at *12:50pm* and transitioned to *Location B at 04:10pm*. From this recorded data we interpolate the data as follows:

- 12:50pm Location A; Duration 0 minutes
- 01:00pm Location A; Duration 10 minutes
- …
- 04:00pm Location A; Duration 190 minutes
- 04:10pm Location B; Duration 0 minutes

Transition Time Prediction

In order to predict when a person will transition from one location to the next, our algorithm answers the question "Will the occupant stay at the current location for the next *n* minutes?" This is realized as a binary prediction with 1 representing *yes* and 0 representing *no*. Recall that the algorithm uses an ensemble method. For this prediction, this consists of a simple mean of the predictions of the four classifiers. If the mean prediction returned by the ensemble method >= 0.5, Indoor-ALPS assigns 1 as a prediction result, and it assigns 0 otherwise.

However, as different applications are likely to require predictions about different stay durations, we trained algorithms and performed predictions for nine specific time windows: n = 10, 20, 30, …, 90. This approach can be used to predict the time for transitioning from the current location. We predict stay duration for successively larger times, and for the time *t* when the ensemble classifier returns a 0, we infer that the user will leave only after that time.

Prediction of Next Location

The prediction of the next significant location works similarly to the prediction of the transition time. Again using the algorithm described above we ask the question "What is the next significant location the user will transition to?". Unlike the binary prediction of staying in the current location for a particular time window, here we are solving a multiclass problem, where the classes are the set of significant locations. We use the same interpolated data we created for the temporal prediction and record the next location for each data point. For example in the data snippet above, each interpolated data point has Location A as the next location.

For this prediction, our ensemble classifier uses a majority voting approach. Each algorithm makes a prediction and the most common prediction is used, rather than the average as was used with the transition time prediction. If there is no clear majority, we choose the location at random from among the predicted locations.

Temporal-Spatial Prediction

To use the two independent predictions for temporal and spatial prediction we chain them together as follows: first we run the temporal prediction to determine if the occupant will stay at the current location or not, and secondly, if the occupant is predicted to leave, we run the spatial prediction to determine where she will transition to. We now describe the results from applying our algorithm to a real-world dataset.

INDOOR-ALPS EVALUATION

In order to evaluate and better understand the performance of Indoor-ALPS, we compared it to Prediction by Partial Match (PPM) [3], a state of the art algorithm for predicting occupancy and location transition. We selected a publicly available data set, the Augsburg Indoor Location Tracking Benchmark [15] to compare the algorithms.

Using Prediction by Partial Match as a Benchmark

Prediction by Partial Match (PPM) [3], to our knowledge, has the highest reported accuracy for the task of predicting when an occupant transitions from one location to the next and where the occupant transitions to. PPM is based on a 1st order Markov Model. The input data for this algorithm is formatted into 10-minute time slots and for each time slot PPM calculates the frequency that each location was recorded for that time slot. In order to make a prediction, the algorithm takes a time slot as an input and returns the location with the highest frequency.

We used PPM to make both temporal and spatial predictions. As we explained previously, PPM takes a discrete timestamp and predicts the most frequent location for that timestamp. To make *temporal* predictions, given a current time index $\boldsymbol{t}$ and look-ahead window $\boldsymbol{n}$ (to answer the question: "Are you staying at this location for the next $\boldsymbol{n}$ minutes?"), we query PPM to predict the location for the next $\boldsymbol{t+i}$ (i=1,..,$\boldsymbol{n}$) time slots. If the resulting location for any of these $\boldsymbol{n}$ predictions is different from the current one, we assign 0 as the result (person is not staying for the next $\boldsymbol{n}$ minutes) or 1 otherwise. In order to make *spatial* predictions given a current time index $\boldsymbol{t}$, we query PPM to predict the location for the next $\boldsymbol{t+i}$ (i=1,..,143, corresponding to a one-day look-ahead) time slots. The first predicted location that is different from the current one is returned as the next location.

Evaluation Data Set

The Augsburg data set has been used extensively in previous attempts to predict location [14,17,19,23]. We chose it for our analysis as it enables easier comparison of our algorithm to past and future efforts to address indoor location prediction. The data set was collected using a smart doorplate concept: each user was tracked using an RFID card upon entering a space (*e.g.*, office, corridor, kitchen). Four university office workers who worked on the same floor were tracked for an average of 6.75 weeks (SD=1.71). Due to the nature of the data collection, this data set already contains distinct locations, thus preprocessing is not needed to extract them. However, as we care about significant locations [1], we performed a preprocessing step to filter out data where the occupant left a room for less then 10 minutes and returned to the same room.

Measures

To give a complete picture of the performance of each algorithm, we report on four different performance measures: Accuracy, Precision, Recall, and Kappa statistic.

Accuracy

The accuracy for an algorithm describes the fraction of correct predictions vs. total predictions. We calculated the accuracy for each individual user and look-ahead window and averaged the resulting values. The accuracy is closely tied to the underlying distribution of the different classes, meaning a dominant class that is predicted very well will raise the overall accuracy.

Precision and Recall

Precision and Recall are information retrieval concepts that provide an understanding and measure of relevance and it quantifies the false positive and false negative errors being made by an algorithm. For a given class, *Precision* describes the fraction of data points that are predicted to belong to that class and actually belong to that class. *Recall*, on the other hand, describes how many data points that belong to a class are actually retrieved. Precision is tied to the false positive value and Recall is tied to the false negative value. Both precision and recall are calculated on a per-class basis for each user in the data set. The overall precision and recall for an algorithm on a specific data set is calculated by using a weighted average of the class specific precision and recall. The number of data points belonging to a particular class is used as the weight for this average. Where accuracy shows the overall correctness of an algorithm, precision and recall tell us how an algorithm achieved a certain accuracy value and the kind of errors it makes. Depending on the application, false positives or false negatives may be more of a concern.

Kappa Statistic

The Kappa statistic measures the class-wise agreement between actual data and predicted data. For each observed class, it calculates how many data points were correctly predicted to be in that class and how many data points were incorrectly predicted to be in any of the other possible classes. Kappa evaluates how well an algorithm is able to predict the individual classes in a dataset independent of the class distribution. For example, if an algorithm is very well suited to predict the two most common classes, but is unable to predict any of the other classes, Kappa would be very low. Thus it is possible to have high accuracy, precision, and recall values, but a low Kappa. In our analysis, we apply the commonly used Landis and Koch [10] to interpret our Kappa results.

RESULTS

We split the analysis of our proposed approach into three parts: temporal, spatial, and combined temporal-spatial. We performed a statistical analysis on the accuracy of each algorithm. We then report other measures to support the discussion about the advantages and limitations of each individual algorithm.

In all of our statistical analysis we used One-way and Two-way Repeated Measures ANOVAs, as appropriate. We used Repeated Measures ANOVAs because algorithms were trained on data for each individual user, and the training and testing data split for both algorithms was the same. We ensured the normality of data using the Shapiro-Wilk test. We ran Mauchley's test for sphericity and performed Greenhouse-Geisser correction when the sphericity assumption was violated. All *post-hoc* pair-wise comparisons were done using paired t-tests with Bonfferoni correction. We report only main effects and p-values in the text for succinctness.

Temporal Analysis

Different applications require different temporal look-ahead windows. For example, the look-ahead window for indoor temperature control is based on the average time it takes to raise or lower the temperature in a room to a comfortable temperature, and this can vary based on the current time of day, outside temperature, and position of the room in the building (is the room inside of the building, does it have windows, *etc.*). To show how Indoor-ALPS performs under different temporal thresholds we report the results for nine different look-ahead windows, from a 10-minute to a 90-minute look-ahead. We compare our results against PPM and a 0-R predictor (majority class predictor).

The ground truth for the temporal prediction is tied to the look-ahead window and consists of a series of 0's and 1's. Given a look-ahead window *n* and a data point, the ground truth for that data point is 1 if the person stays at the current location for the next *n* minutes. If, on the other hand, the remaining stay-duration is smaller than *n*, we assign 0 for the ground truth.

We expect that the 0-R predictor has very high accuracy for lower look-ahead windows, since the only instances when an occupant is predicted to leave a location is shortly before the transition time. This makes the prediction task especially difficult for lower thresholds, because there is little data available for the *leave*-class.

Figure 1 shows the average accuracy across all four users for all 9 look-ahead windows by algorithm. Our tests found a main effect of the *Algorithm* on the *Accuracy* ($F_{(1,3.01)}$=11.56, p=.0422, η_p^2=.79); Indoor-ALPS overall mean accuracy of 88.2% was significantly higher than PPM (mean=83.6%, p<.0001) and 0-R (mean=83.3%, p<.0001). Our tests did not find a significant difference between PPM and 0-R (p>.9999).

Our tests also found a main effect of *Look-Ahead* on *Accuracy* ($F_{(1.13,3.39)}$=144.07, p=.0006, η_p^2=.97). As the look-ahead increased, the accuracy of all three algorithms decreased. This makes sense, as it is harder to predict mobility further into the future. Our tests also found simple effects of *Algorithm* on *Accuracy* for all individual *Look-Aheads*. In particular, Indoor-ALPS was significantly more accurate than 0-R and PPM at the 10 minute, 20 minute, 30 minute and 70 minute look-aheads (See Figure 2, p<=0025 in all cases).

Thus, Indoor-ALPS performs better overall than PPM and 0-R, and is especially better for small look-ahead windows of 10 to 30 minutes. The average accuracy difference between Indoor-ALPS and PPM for the first 3 look-ahead windows is 9.2% (SD=2.3%) and 3.9% (SD=0.2%) for the higher look-ahead windows.

Figure 2 shows Precision and Recall for all look-ahead windows for Indoor-ALPS as well as PPM. Note by definition, 0-R will always have a Recall of 0. We see a slight improvement for the Precision of Indoor-ALPS over PPM for the first five look-ahead windows after which PPM has a slightly higher Precision. For the smaller look-aheads, PPM makes more false positive errors, *i.e.*, predicting that an occupant is staying when they are not. For the larger look-aheads, Indoor-ALPS makes more false positive errors. However, Indoor-ALPS has a consistently higher Recall than PPM with an average Recall across all look-aheads of 96.6% (SD=2.1%) for Indoor-ALPS and 83.6% (SD=2.9%) for PPM. PPM's lower Recall means that it makes more false negative errors, predicting that an occupant is leaving a location a lot earlier than they actually do.

Figure 3 shows the Kappa results by look-ahead window and algorithm. As expected from the previous Accuracy, Precision and Recall results, Indoor-ALPS has higher Kappa values for the lower look-ahead windows and slightly worse Kappa values for the higher look-ahead windows. The Kappa values for PPM increase as the look-ahead gets larger. It is also of note that the standard deviation for the Kappa values is fairly uniform for PPM (avg.=0.075, SD=0.013), while the standard deviation increases for higher look-ahead windows for Indoor-ALPS (avg.=0.087, SD=0.052). In the discussion section, we will provide a rationale for why this occurred.

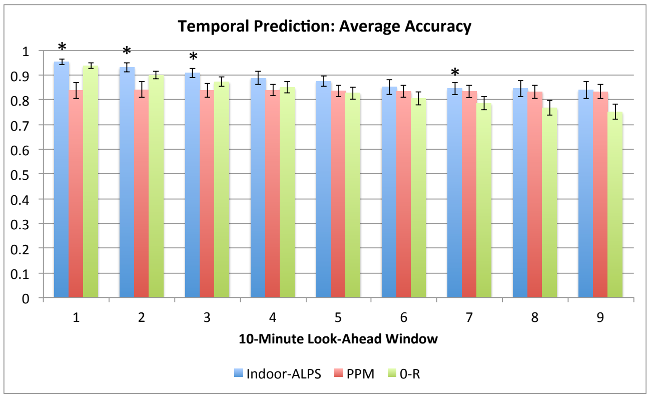

Figure 1 Average Accuracy by Look-Ahead Window for Temporal Prediction. Starred bars indicate statistically higher values (p<=.025).

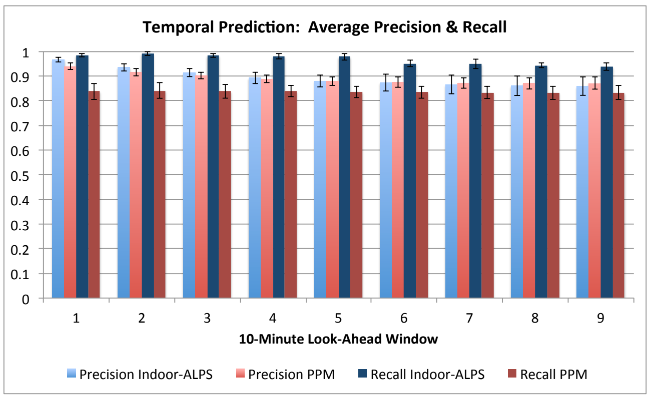

Figure 2 Average Precision & Recall for Temporal Prediction

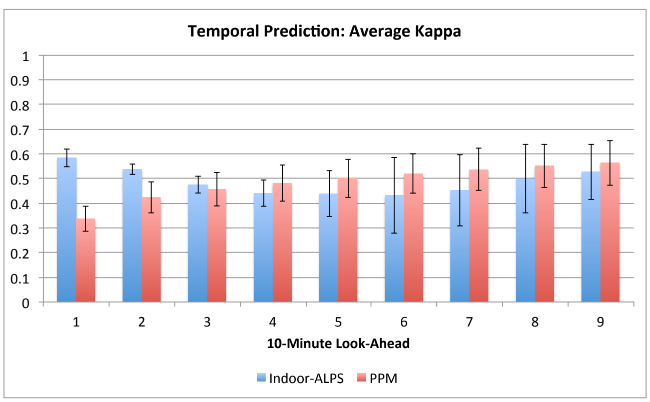

Figure 3 Average Kappa for Temporal Prediction

Spatial Analysis

In order to evaluate the performance of the spatial or next significant location prediction, we again report on the Accuracy, Precision, Recall, and Kappa statistic. The ground truth for this analysis is the next significant location that follows the current location. Figure 4 shows the accuracy by user for each algorithm. As we can see, both Indoor-ALPS as well as PPM outperform 0-R. The accuracies for Indoor-ALPS and PPM are quite similar, with PPM outperforming Indoor-ALPS for User 3. Note that we cannot run statistical tests on accuracy across our users for our spatial analysis as we only have 4 users and thus only 4 values per algorithm. We also analyzed Precision and Recall (see Figure 5) and found a similar picture as with the accuracy. Looking at Kappa (see Figure 6) the algorithms again have similar performance. We can see that both Indoor-ALPS and PPM have a substantial agreement between the observed and predicted data for users 1 and 2, a moderate agreement for user 3, and Indoor-ALPS has a slight agreement and PPM has a fair agreement for user 4. Overall, we can draw the conclusion that PPM offers modest improvements over Indoor-ALPS for users 3 and 4, and performs similarly for users 1 and 2.

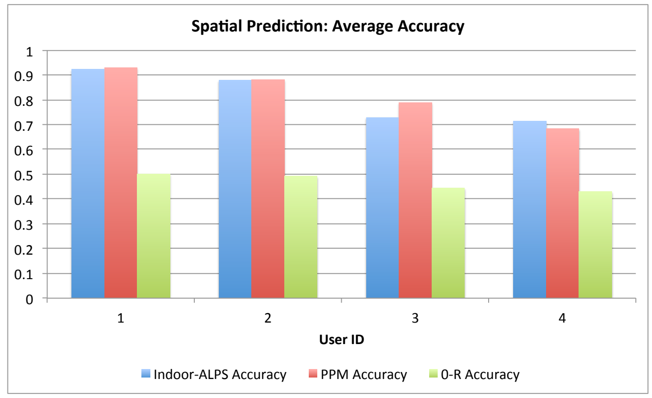

Figure 4 Accuracy by User ID for Spatial Prediction

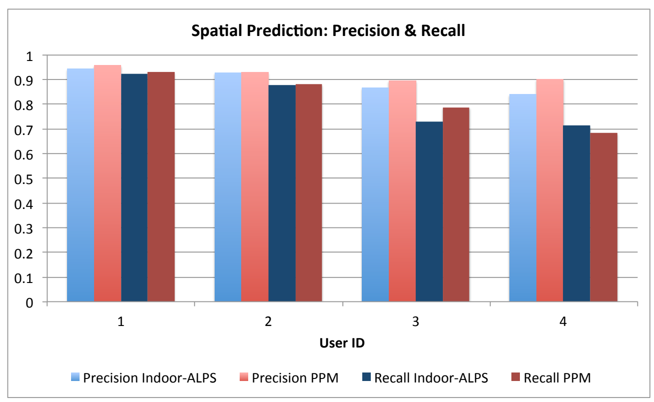

Figure 5 Precision & Recall by User ID for Spatial Prediction

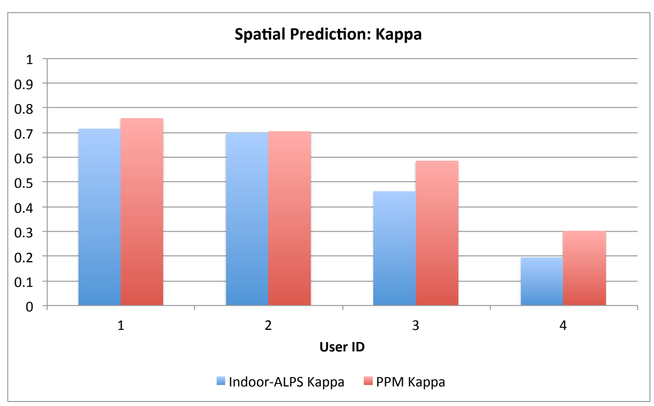

Figure 6 Kappa by User ID for Spatial Prediction

Temporal-Spatial Analysis

Our algorithm combines the results of the temporal and spatial prediction to predict if an occupant will stay at a location for a given time duration, and if the occupant is predicted to leave, to which location the occupant is transitioning. To evaluate the algorithms, we first calculated how often they correctly predict that an occupant is staying, and when they correctly predict that the occupant is leaving, how often they correctly predict the next location. Figure 7 shows the average prediction accuracy by look-ahead window. Note we did not include 0-R in this analysis as the results are identical to the temporal 0-R, since 0-R predicts that the user is always staying and thus no location prediction is performed.

Our tests found a main effect of *Algorithm* on *Accuracy* ($F_{(1,3)}$=65.65, *p*=.0038, η_p^2=.96), indicating that Indoor-ALPS (mean=85.3%) overall performs better than PPM (mean=79.1%). Our tests also found a main effect of *Look-Ahead* on the *Accuracy* ($F_{(8,24)}$=66.75, *p*<.0001, η_p^2=.96), but the *post-hoc* pairwise comparison did not find any significant difference between *Look-Aheads*.

The analysis of the algorithms and look-ahead windows also showed a significant interaction between *Algorithm* and *Look-Ahead* ($F_{(1.75,5.26)}$=15.58, *p*=.0067, η_p^2=.84). Again, we only further compare algorithms across different look-ahead windows. Indoor-ALPS was significantly more accurate than PPM for look-ahead windows of 10 to 80 minutes (all *p*<.05) (Figure 7). Our tests only failed to find a significant difference for the look-ahead window of 90 minutes (*p*=.63). The greatest difference can be observed over the first three windows with an average difference of 10.7% (SD=2.2%), with a maximum improvement of 12.9%. For the first five windows, it is 9.1% (SD=2.7%). This means that combining temporal and spatial prediction in Indoor-ALPS is more accurate than PPM overall, and although the performance difference gets smaller as the look-ahead grows, Indoor-ALPS performs better than PPM in almost all look-ahead windows.

Figure 8 shows the Precision and Recall results for each algorithm. We can see that Indoor-ALPS is slightly worse on Precision with an average of 0.852 (SD=0.051) compared to 0.864 (SD=0.049) for PPM, but it is much better for Recall with an average of 0.915 (SD=0.039) for Indoor-ALPS and 0.794 (SD=0.010) for PPM. Indoor-ALPS' Recall is on average 0.121 (SD=0.032) greater than PPM's Recall, which only slightly changes from one look-ahead window to the next. The lower Recall indicates that PPM leads to more false negatives.

Compared to PPM, Indoor-ALPS has a higher Kappa (see Figure 9) for the lower look-ahead windows and a slightly lower Kappa for the higher look-ahead windows. Similar to the temporal prediction, the standard deviation for Indoor-ALPS' Kappa increases as the look-ahead grows, while the standard deviation for PPM's Kappa is fairly stable with an average of 0.051 (SD=0.006).

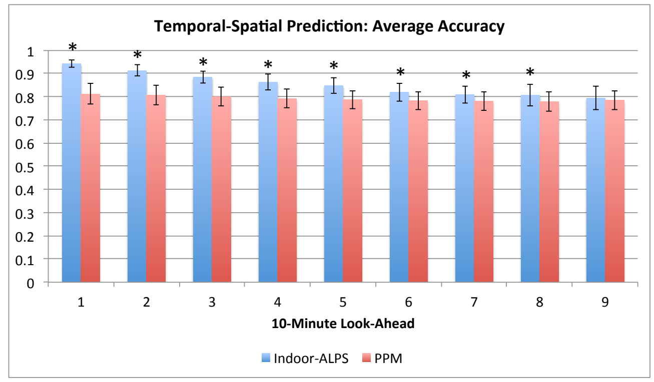

Figure 7 Average Accuracy by Look-Ahead for Temporal-Spatial Prediction. Star indicates statistically significant difference (p<.05) .

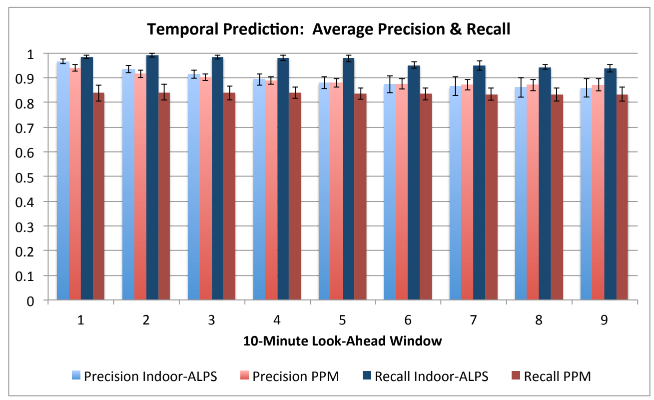

Figure 8 Average Precision & Recall by Look-Ahead Window for Temporal-Spatial Prediction.

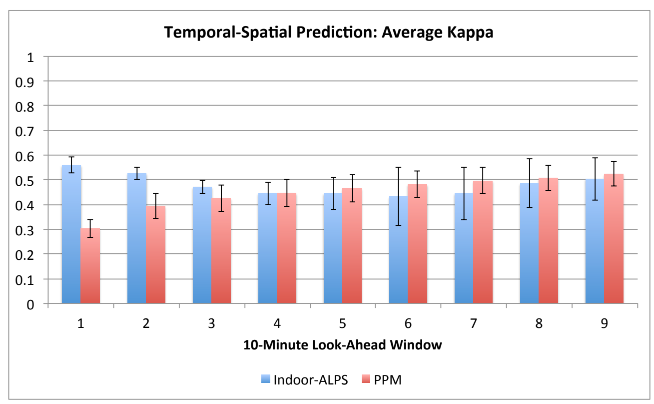

Figure 9 Average Kappa by Look-Ahead Window for Temporal-Spatial Prediction.

DISCUSSION

Our performance analysis of Indoor-ALPS compared to PPM shows that Indoor-ALPS achieves higher overall temporal and temporal-spatial prediction performance. Furthermore, Indoor-ALPS performs particularly well for the look-ahead time windows of 10 to 30 minutes. Especially for our main prediction goal, temporal-spatial, it outperforms PPM for all window sizes up to 80 minutes in terms of accuracy. This discussion will highlight the advantages and disadvantages of Indoor-ALPS and explain why PPM performs better in some cases. We also illustrate how Indoor-ALPS can be used in practice in an application.

Temporal Prediction

As we described in the results section, Indoor-ALPS achieves a very high performance increase over PPM on multiple measures for the temporal prediction. This is especially true for lower look-ahead windows for which Indoor-ALPS had a mean accuracy gain of 7.3% (SD=3.1%) over PPM. Furthermore our algorithm also improves over 0-R even for lower look-ahead windows for which the 0-R accuracy is already very high.

However, the Kappa results for higher look-ahead windows were worse than PPM. As highlighted earlier, we observed a higher standard deviation for the higher look-aheads for Indoor-ALPS. This indicates that there were significant differences in the performance of our algorithm for one or more users. Careful analysis of the results shows that user 3 was responsible for these deviations. In fact, if we remove this user from the analysis and recalculate the average Kappa for both Indoor-ALPS and PPM for look-ahead windows of 50 to 90 minutes, we see that Kappa increases by 0.057 for Indoor-ALPS while it only increases by 0.015 for PPM when compared to the results for all four users. An analysis of user 3's data revealed that in the validation data, the user frequently went to two additional new locations, which were not present in the initial training data. If we analyze the selected features of Indoor-ALPS for each validation day and look-ahead window we notice that the current location is one of the most frequent features picked by the feature selection algorithm. Since the data for these new locations is relatively sparse in the training data, our algorithm frequently predicts that the occupant stays. Even with incremental learning, it takes a while to collect enough data about the new location to make accurate predictions. This is also reflected in the Precision, which is lower than the average precision of users 1, 2, and 4 by 0.055.

PPM on the other hand is agnostic to the current location when it makes a prediction for a given time slot. It only uses discrete time to make a prediction. As long as the predicted location differs from the current one it will predict that the occupant leaves within a given look-ahead window. This is why PPM is more robust to changes in a person's routine. For our future work, we plan to extend Indoor-ALPS and allow it to react to changes in the user's routine by falling back to a frequency-based model.

Even without including user 3 in the results we still see that Kappa is slightly higher for PPM on look-aheads of 40 to 70 minutes (average difference of 0.042). This behavior can be explained by looking at the Recall for PPM. We see that PPM has a much lower Recall, indicating that the algorithm very frequently predicts that the user will leave her location even if she actually stays (False Negative errors). By doing so, it correctly predicts more of the Leave-class, which is the minority class in the data set, but at the expense of the Stay-class, resulting in a slightly higher Kappa. The slightly lower Precision shows that Indoor-ALPS makes the opposite errors; when the data shows that the occupant leaves within the next *n* minutes, our algorithm sometimes incorrectly predicts that the occupant will stay.

Spatial Prediction

The evaluation of the spatial prediction showed that Indoor-ALPS and PPM outperform 0-R and both achieve comparable performance. Only for user 3 we see that PPM achieves a higher accuracy than Indoor-ALPS, with a difference of 5.9% between the two. The cause for this is the same as was described for the temporal prediction for user 3. Since the prediction algorithm uses the current location as a feature, it has difficulties with two new locations. The locations that are transitioned to when leaving these two new locations are very frequented (or majority) locations for this user, which is why PPM is better able to handle the new situation since it predicts the majority location. One potential way to improve the next location prediction is by leveraging the temporal prediction along with Active Learning. In situations when the temporal prediction predicts that the occupant is leaving the current location and the prediction is uncertain about the most likely next location, it can ask the user for input to improve its ability to learn.

Temporal-Spatial Prediction

Analyzing the overall performance of the temporal-spatial prediction, the task we set out to solve with Indoor-ALPS, we can see that our algorithm is significantly more accurate than PPM, where it performs much better particularly on the lower look-ahead windows. Our proposed algorithm addresses the following problem: predicting if an occupant will stay at their current location for a specified time frame and, if they are predicted to leave, predicting also the indoor location they will transition to within that time frame. Our algorithm with its high prediction performance can support a large class of compelling applications, including proactive traffic notifications, proactive heating, automated receptionists, and *ad-hoc* meeting support. We now look in detail at how Indoor-ALPS supports proactive heating.

Using Indoor-ALPS for Proactive Heating

The importance of the different look-ahead windows is dependent on the application context. For example, efficient temperature control in domestic environments requires a look-ahead of, on average, 60 minutes [8] to change temperature by 10°F. In office environments with zoned temperature control, the heat-up time is usually smaller due to the smaller volume of the space that is heated and secondary heating effects from adjacent rooms. A look-ahead window of 30 minutes is already enough to positively affect a building's overall energy consumption while minimizing the impact on the occupant's thermal comfort. Our algorithm is particularly suited to solve this problem.

Let us assume the HVAC system in a building needs 30 minutes to change the temperature by 10°F, which requires a 30 minute look-ahead window for knowing if a person will transition to a new space. Given a particular location, our algorithm would make temporal predictions every 10 minutes in order to determine if the person is staying at their current location for at least 30 minutes. As soon as the prediction result changes from yes to no in consecutive queries, Indoor-ALPS makes a spatial prediction to identify which location to heat up. In this case, the HVAC system can be controlled to start increasing the temperature to the person's preferred temperature in advance of his arrival.

As we saw in the temporal-spatial results, we are not 100% accurate in our prediction, which is to be expected. There are two possible errors that can occur, which would affect the temperature control for the person's next location: 1) the spatial prediction was incorrect and the person is heading to a different office or 2) the spatial prediction was correct, but the temporal prediction was not. In case of a wrong spatial prediction the system would waste energy heating the wrong room. In case of a correct temporal prediction, heating will start on time or a little bit late, since we expect they may leave anytime in the next 30 minutes.

In the 11.5% of cases where an error occurs, two types are possible: predictions that the person will leave when they actually do not leave the current location in the next 30 minutes or the prediction that the person will stay when they actually will leave in the next 30 minutes. The Precision and Recall results for Indoor-ALPS and PPM have shown that Indoor-ALPS is more likely to make the latter error, while PPM is more likely to make the former error. Thus the primary impact of temporal errors with Indoor-ALPS will be a delay in heating.

We believe that even though our algorithm sometimes starts heating too late, it will proactively reduce the energy consumption of a building. The errors Indoor-ALPS makes might affect the thermal comfort of the occupant, because the temperature did not reach the preferred temperature on time or in the correct room. The difference from the preferred temperature due to slight prediction inaccuracies might not even affect the thermal comfort much since the system only takes 10 to 15 minutes to recover and someone who was just walking is less likely to be cold. Looking at the large Recall difference between Indoor-ALPS and PPM we can expect that PPM would waste more energy than our algorithm and thus Indoor-ALPS would be better suited to reducing a building's energy footprint.

Following the same prediction model as described above another application our algorithm can support is an automated receptionist [5]. By predicting that an office occupant will remain in her office for a certain amount of time or longer, a colleague or visitor can be assured that she will be there when they arrive. Here the look-aheads are likely to be short, where Indoor-ALPS excels. By predicting if the building occupant will arrive at a new location in a specified amount of time (or less), we can assure the visitor he will meet the occupant if he arrives around that time.

Combining Temporal and Spatial Predictions with Oracles

Even though Indoor-ALPS already achieved a very high performance, we were interested to see whether we could further improve its performance by combining the temporal and spatial prediction (*i.e.*, allow the output of one prediction become the input for the other). To test this approach, we created two Oracle predictors. The Temporal-Oracle predictor has perfect knowledge about the next location of an occupant and uses that knowledge as an input feature in the prediction. The Spatial-Oracle, on the other hand, has perfect knowledge about if an occupant stays for the next n (n=10,…,90) minutes. If either Oracle with perfect knowledge (either spatially or temporally) results in a significantly improved performance, then we can try combining our imperfect predictors. However, when we evaluated the use of these Oracles, we found that neither one provided an improvement over our original approach. Note that this result is true for the Augsburg dataset, and may still be worth investigating with a different dataset containing different types of mobility patterns.

CONCLUSION

In this paper we presented Indoor-ALPS, an Adaptive Indoor Location Prediction System, which predicts if a person will stay at their current location or if she is predicted to leave, to which location she will transition to. Its novelty is in treating the temporal and spatial aspects of the problem as independent and then combining the results. We implemented Indoor-ALPS and tested it on the Augsburg Indoor Location Tracking Benchmark against the best performing algorithm for this prediction problem, Prediction by Partial Match. Our analysis showed that Indoor-ALPS improved on the temporal-spatial accuracy by 6.2% overall, and 10.7% over PPM on lower look-ahead windows. These positive results will allow us to realize various applications such as zoned temperature control, opportunistic meetings, or proactive traffic notifications. For future work we plan to evaluate our algorithm on other datasets to demonstrate the generalizability of our results. In addition we also plan to adopt our temporal prediction for other indoor prediction tasks such as room occupancy prediction. To improve our algorithm's performance, we plan to explore the integration of frequency-based models in situations when the prediction for a partial day is consistently wrong for multiple look-aheads. In order to make Indoor-ALPS more suitable for long-term deployment we plan to include a re-learning rate that allows it to allows it discard old data in favor of new data, which will make it more robust against changes in a person's temporal-spatial routine. Lastly we plan to evaluate Indoor-ALPS by building and deploying the applications that the algorithms support.

ACKNOWLEDGEMENTS

This work was supported in part by NSF grants IIS-1227495 and IIS-1217929, and the Portuguese Foundation for Science and Technology SFRH/BD/70428/2010.

REFERENCES

1. Ashbrook, D., & Starner, T. (2003). Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, *7*(5), 275-286. Springer.
2. Balaji, B., Xu, J., Nwokafor, A., Gupta, R., & Agarwal, Y. (2013, November). Sentinel: occupancy based HVAC actuation using existing WiFi infrastructure within commercial buildings. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems* (article 17). ACM.
3. Burbey, I. E. (2011). *Predicting future locations and arrival times of individuals* (Doctoral dissertation, Virginia Polytechnic Institute and State University).
4. Ellis, C., Scott, J., Hazas, M., & Krumm, J. (2012). Earlyoff: Using house cooling rates to save energy. In *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings* (pp. 39-41). ACM.
5. Gockley, R., Bruce, A., Forlizzi, J., Michalowski, M., Mundell, A., Rosenthal, S., Sellner, B., Simmons, R., Snipes, K., Schultz, A.C., & Wang, J. (2005). Designing robots for long-term social interaction. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on* (pp. 1338-1343). IEEE.
6. Gomes, R., Welling, M., & Perona, P. (2008). Incremental learning of nonparametric Bayesian mixture models. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on (pp. 1-8). IEEE.
7. Jain, S., Lange, S., Zilles, S. (2006). Towards a better understanding of incremental learning. In *Algorithmic Learning Theory* (pp. 169-183). Springer Berlin Heidelberg,
8. Koehler, C., Ziebart, B. D., Mankoff, J., & Dey, A. K. (2013). TherML: occupancy prediction for thermostat control. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing* (pp. 103-112). ACM.
9. Krumm, J., & Horvitz, E. (2006). Predestination: Inferring destinations from partial trajectories. In *UbiComp 2006: Ubiquitous Computing* (pp. 243-260). Springer Berlin Heidelberg.
10. Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *biometrics*, *33*(1), 159-174.
11. Lee, K. C., & Cho, H. (2010). Performance of Ensemble Classifier for Location Prediction Task: Emphasis on Markov Blanket Perspective. *International Journal of U-& E-Service, Science & Technology*, *3*(3).
12. Lee, S., Lee, K. C., & Cho, H. (2010). A Dynamic Bayesian Network Approach to Location Prediction in Ubiquitous Computing Environments. In *Advances in Information Technology* (pp. 73-82). Springer Berlin Heidelberg.
13. Mynatt, E., & Tullio, J. (2001). Inferring calendar event attendance. In *Proceedings of the 6th international conference on Intelligent user interfaces* (pp. 121-128). ACM.
14. Oh, S. (2012). Using an Adaptive Search Tree to Predict User Location. *Journal of Information Processing Systems*, *8*(3).
15. Petzold, J. (2004, February) Augsburg Indoor Location Tracking Benchmarks. *Technical Report 2004-9*, Institute of Computer Science, University of Augsburg, Germany, February 2004. http://www.informatik.uni-augsburg.de/skripts/techreports/
16. Petzold, J., Bagci, F., Trumler, W., & Ungerer, T. (2003). Global and local state context prediction. In *Artificial Intelligence in Mobile Systems*.
17. Petzold, J., Bagci, F., Trumler, W., & Ungerer, T. (2006). Comparison of different methods for next location prediction. In *Euro-Par 2006 Parallel Processing* (pp. 909-918). Springer Berlin Heidelberg.
18. Pudil, P., Novovičová, J., & Kittler, J. (1994). Floating search methods in feature selection. *Pattern recognition letters*, *15*(11), 1119-1125.
19. Ryan, C., & Brown, K. N. (2012). Occupant Location Prediction Using Association Rule Mining. In *Workshop on AI Problems and Approaches for Intelligent Environments* (p. 27).
20. Scott, J., Bernheim Brush, A. J., Krumm, J., Meyers, B., Hazas, M., Hodges, S., & Villar, N. (2011). PreHeat: controlling home heating using occupancy prediction. In *Proceedings of the 13th international conference on Ubiquitous computing* (pp. 281-290). ACM.
21. U.S. Census Bureau. (2009) *Commuting in the United States* (http://www.census.gov/prod/2011pubs/acs-15.pdf)
22. Vintan, L., Gellert, A., Petzold, J., & Ungerer, T. (2004). Person movement prediction using neural networks. In *First Workshop on Modeling and Retrieval of Context*.
23. Vuong, N. K., Chan, S., Lau, C. T., & Lau, K. M. (2011). A predictive location-aware algorithm for dementia care. In *Consumer Electronics (ISCE), 2011 IEEE 15th International Symposium on* (pp. 339-342). IEEE.
24. Ziebart, B. D., Maas, A. L., Dey, A. K., & Bagnell, J. A. (2008). Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior. In *Proceedings of the 10th international conference on Ubiquitous computing* (pp. 322-331). ACM.