Published in final edited form as:
IEEE Trans Cybern. 2013 June ; 43(3): 820–828. doi:10.1109/TSMCB.2012.2216873.

Activity Discovery and Activity Recognition: A New Partnership

Diane Cook, Fellow, IEEE,
School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA, 99163

Narayanan Krishnan, Member, IEEE, and
School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA, 99163

Parisa Rashidi, Member, IEEE
Computer and Information Science and Engineering Department, University of Florida, Gainesville, FL, 32611

Abstract

Activity recognition has received increasing attention from the machine learning community. Of particular interest is the ability to recognize activities in real time from streaming data, but this presents a number of challenges not faced by traditional offline approaches. Among these challenges is handling the large amount of data that does not belong to a predefined class. In this paper, we describe a method by which activity discovery can be used to identify behavioral patterns in observational data. Discovering patterns in the data that does not belong to a predefined class aids in understanding this data and segmenting it into learnable classes. We demonstrate that activity discovery not only sheds light on behavioral patterns, but it can also boost the performance of recognition algorithms. We introduce this partnership between activity discovery and online activity recognition in the context of the CASAS smart home project and validate our approach using CASAS datasets.

Index Terms

sequence discovery; activity recognition; out of vocabulary detection

1 Introduction

The machine learning and pervasive computing technologies developed in the last decade offer unprecedented opportunities to provide ubiquitous and context-aware services to individuals. In response to these emerging opportunities, researchers have designed a variety of approaches to model and recognize activities. The process of discerning relevant activity information from sensor streams is a non-trivial task and introduces many difficulties for traditional machine learning algorithms. These difficulties include spatio-temporal variations in activity patterns, sparse occurrences for some activities, and the prevalence of sensor data that does not fall into predefined activity classes.

One application that makes use of activity recognition is health-assistive smart homes and smart environments. To function independently at home, individuals need to be able to complete Activities of Daily Living (ADLs) [1] such as eating, dressing, cooking, drinking, and taking medicine. Automating the recognition of activities is an important step toward monitoring the functional health of a smart home resident [2], [3], [4] and intervening to improve their functional independence [5], [6].

The generally accepted approach to activity recognition is to design and/or use machine learning techniques to map a sequence of sensor events to a corresponding activity label. Online activity recognition, or recognizing activities in real time from streaming data, introduces challenges that do not occur in the case of offline learning with pre-segmented data. One of these challenges is recognizing, and labeling or discarding, data that does not belong to any of the targeted activity classes. Such "out of vocabulary" detection is difficult in the context of activity recognition, and is particularly challenging when the out of vocabulary data represents a majority of the data that is observed.

In this paper we introduce an unsupervised method of discovering activities from sensor data. The unsupervised nature of our approach provides a method of analyzing data that does not belong to a predefined class. By modeling and tracking occurrences of these patterns alongside predefined activities, the combined approach can also boost the performance of activity recognition for the predefined activities.

Here we introduce our approaches to online activity recognition, activity discovery, and our discovery-based boosting of activity recognition. We evaluate the effectiveness of our algorithms using sensor data collected from three smart apartments while the residents of the apartment live in the space and perform their normal daily routines.

2 Datasets

We treat a smart environment as an intelligent agent that perceives the state of the residents and the physical surrounding using sensors, and acts on the environment using controllers in such a way that specified performance measures are optimized [7].

To test our ideas, we analyze sensor event datasets collected from three smart apartment testbeds. Figure 1 shows the floorplan and sensor layout for the three apartments and Figure 2 shows occurrences of activities in each of the testbeds for a sample of the data. Each of the smart apartments housed an older adult resident and is equipped with infrared motion detectors and magnetic door sensors. During the six months that we collected data in the apartments, the residents lived in these apartments and performed normal daily routines.

In order to provide ground truth for the activity recognition algorithms, human annotators analyzed a 2D visualization of the sensor events. They tagged sensor event data with the beginning and ending of activity occurrences for the 11 activities listed in Figure 2. Table 1 lists characteristics of these datasets. Note that although there are many occurrences of the activities, only 42% of the sensor events on average belong to one of the predefined activities.

3 Activity Recognition

The goal of activity recognition is to recognize common human activities in real life settings. In terms of a machine learning approach, an algorithm must learn a mapping from observable data (typically a sequence of sensor events) to an activity label. We describe previous work done in this area together with the approach we adopt for online activity recognition.

3.1 Previous Work

Activity recognition is not an untapped area of research. Because the need for activity recognition algorithms is great, researchers have explored a number of approaches to this problem [8]. The approaches can be broadly categorized according to the type of sensor data that is used for classification, the model that is designed to learn activity definitions, and the realism of the environment in which recognition is performed.

Sensor data—Researchers have found that different types of sensor information are effective for classifying different types of activities. When trying to recognize ambulatory movements (e.g., walking, running, sitting, standing, climbing stairs, and falling), data collected from accelerometers positioned on the body has been used [9], [10]. More recent research has tapped into the ability of a smart phone to act as a wearable / carryable sensor with accelerometer and gyroscope capabilities. Researchers have used phones to recognize gestures and motion patterns [11], [12].

For other activities that are not as easily distinguishable by body movement alone, researchers observe an individual's interaction with key objects in the space such as medicine containers, key, and refrigerators [13], [14], [15]. Objects are tagged with shake sensors or RFID tags and are selected based on the activities that will be monitored. Other researchers rely upon environment sensors including motion detectors and door contact sensors to recognize ADL activities that are being performed [16], [17], [18].

For recognition of specialized classes of activities, researchers use more specialized sources of information. As an example, Yang, et al. [19] collected computer usage information to recognize computer-based activities including multiplayer gaming, movie downloading, and music streaming. In addition, some researchers such as Brdiczka et al. [20] video tape smart home residents and process the video to recognize activities. Because our study participants are uniformly reluctant to allow video data or to wear sensors, and because object sensors require frequent charging and are not practical in participant homes, our data collection has consisted solely of passive sensors that could be installed in a smart environment.

Activity models—The number of machine learning models that have been used for activity recognition varies as greatly as the number of sensor data types that have been explored. Naive Bayes classifiers have been used with promising results for offline learning of activities [20], [21], [22], [23] when large amounts of sample data are available. Other researchers [17], [9] have employed decision trees to learn logical descriptions of the activities, and still others [24] employ kNNs. Gu et al. [13] take a slightly different approach by looking for emerging frequent sensor sequences that can be associated with activities and can aid with recognition.

An alternative approach that has been explored by a number of research groups is to exploit the representational power of probabilistic graphs. Markov models [21], [25], [26], [18], dynamic Bayes networks [15], and conditional random fields [27], [28] have all been successfully used to recognize activities, even in complex environments. Researchers have found that these probabilistic graphs, along with neural network approaches [29], [26], are quite effective at mapping pre-segmented sensor streams to activity labels.

Recognition Tasks—A third way to look at earlier work on activity recognition is to consider the range of experimental conditions that have been attempted for activity recognition. The most common type of experiment is to ask subjects to perform a set of scripted activities, one at a time, using the selected sensors [20], [29], [12], [15]. In this case the sensor sequences are well segmented, which allows the researchers to focus on the task of mapping sequences to activity labels.

Building on this foundation, researchers have begun looking at increasingly realistic and complex activity recognition tasks. These setups include recognizing activities that are performed with embedded errors [21], with interleaved activities [30], and with concurrent activities performed by multiple residents [31], [32], [18]. The next major step that researchers have pursued is to recognize activities in unscripted settings (e.g., in a smart home while residents perform normal daily routines) [17], [26]. These naturalistic tasks have relied on human annotators to segment, analyze, and label the data. However, they do bring the technology even closer to practical everyday usage. The realism of activity recognition has been brought into sharper focus using tools for automated segmentation [20], [13], for automated selection of objects to tag and monitor [14], and for transfer of learned activities to new environment settings [16].

3.2 Online Activity Recognition Using AR

One feature that distinguishes previous work in activity recognition from the situation we describe in this paper is the need to perform continuous activity recognition from streaming data, even when not all of the data fits any of the activity classes. In order to perform activity recognition from streaming sensor data, the data cannot be segmented into separate sensor streams for different activities. Instead, we adopt the approach of moving a sliding window over the sensor event stream and identifying the activity that corresponds to the most recent event in the window. This sliding window approach has been used in other work [30], but not yet for activity recognition in unscripted settings. In this study we consider data collected from environmental sensors such as motion and door sensors, but other types of sensors could be included in these approaches as well.

We experimented with a number of machine learning models that could be applied to this task, including naive Bayes, hidden Markov models, conditional random fields, and support vector machines. These approaches are considered for this task because they traditionally are robust in the presence of a moderate amount of noise and are designed to handle sequential data. Among these three choices there is no clear best model to employ - they each utilize methods that offer strengths and weaknesses for the task at hand.

The naive Bayes (NB) classifier uses relative frequencies of feature values as well as the frequency of activity labels found in sample training data to learn a mapping from activity features, D, to an activity label, a, calculated using the formula $argmax_{a \in A} P(a/D) = P(D/a)P(a)/P(D)$. In contrast, the hidden Markov model (HMM) is a statistical approach in which the underlying model is a stochastic Markovian process that is not observable (i.e., hidden) which can be observed through other processes that produce the sequence of observed features. In our HMM we let the hidden nodes represent activities and the observable nodes represent combinations of feature values. The probabilistic relationships between hidden nodes and observable nodes and the probabilistic transitions between hidden nodes are estimated by the relative frequency with which these relationships occur in the sample data.

Like the hidden Markov model, the conditional random field (CRF) model makes use of transition likelihoods between states as well as emission likelihoods between activity states and observable states to output a label for the current data point. The CRF learns a label sequence which correpsonds to the observed sequence of features. Unlike the hidden markov model, weights are applied to each of the transition and emission features. These weights are learned through an expectation maximization process based on the training data.

Our last approach employs support vector machines (SVMs) to model activities. Support vector machines identifies class boundaries that maximize the size of the gap between the boundary and data points. We employ a one vs one support vector machine paradigm that is computationally efficient when learning multiple classes with possible imbalance in the amount of available training data for each class. For the experiments reported in this paper we used the libSVM implementation of Chang et al [33].

We compared the performance of these machine learning models on our real-world smart home datasets. Table 2 summarizes recognition accuracy based on threefold cross validation over each of the real-world datasets. As shown in the table, all of the algorithms perform well at recognizing the 10 predefined activities listed in Figure 2. Although they perform well for these predefined activity classes, there are slight variances in recognition accuracy. The support vector machine model yield the most consistent performance across the datasets. As a result, we utilize only this approach for modeling and recognizing activities for the experiments described in the rest of this paper.

For real-time labeling of activity data from a window of sensor data, we experimented with a number of window sizes and found that using a window size of 20 sensor events performed best. For this reason we adopt these choices for our algorithm recognition approach, called **AR**. Each input data point is described by a set of features that describes the sensor events in the 20-event window. These features include:

- Number of events triggered by each sensor in the space within the window.
- Time of day of the first and last events in the window (rounded to the nearest hour).
- Timespan of the entire window (rounded to the nearest hour).

The machine learning algorithm learns a mapping from the feature representation of the sensor event sequence to a label that indicates the activity corresponding to the last event in the sequence. The default parameters are used for the support vector machine and the shrinking heuristic is employed. All results are reported based on 3-fold cross validation. We recognize that the models could be fine tuned to yield even greater performance for some cases. We also note that alternative models might perform better in different activity recognition situations. In this paper we commit to using a straightforward model that yields consistently strong performance in order to focus on our main contribution: the role of activity discovery in the activity recognition process.

4 Activity Discovery Using AD

A main contribution of this paper is the introduction of an unsupervised learning algorithm to discover activities in raw sensor event sequence data, which we refer to as **AD**. Here we describe previous work in the area and introduce our method for activity discovery.

4.1 Previous Work

Our approach to activity discovery builds on a rich history of discovery research, including methods for mining frequent sequences [34], [13], mining frequent patterns using regular expressions [35], constraint-based mining [36], mining frequent temporal relationships [37], and frequent-periodic pattern mining [38].

More recent work extends these early approaches to look for more complex patterns. Ruotsalainen et al. [39] design the Gais genetic algorithm to detect interleaved patterns in a unsupervised learning fashion. Other approaches have been proposed to mine discontinuous patterns [40], [41], [42] in different types of sequence datasets and to allow variations in occurrences of the patterns [43]. Huynh et al. [44] explored the use of topic models and LDAs to discovery daily activity patterns in wearable sensor data.

Aspects of these earlier techniques are useful in analyzing sensor sequence data. In addition to finding frequent sequences that allow for variation as some of these others do, we also want for our purposes to identify sequences of sufficient length that may constitute an activity of interest. We are interested in characterizing as much of the sensor data as possible but want to minimize the number of distinct patterns to increase the chance of identifying more abstract activity patterns. We describe our approach to meeting these goals next.

4.2 The AD Algorithm

As with other sequence mining approaches, our AD algorithm searches the space of sensor event sequences in order by increasing length. Because the space of possible sequence patterns is exponential in the size of the input data, AD employs a greedy search approach, similar to what can be found in the Subdue [45] and GBI [46] algorithms for graph-based pattern discovery. Input to the AR discovery algorithm includes the input sensor data set, a beam length, and a specified number of discovery iterations.

AD searches for a sequence pattern that best compresses the input dataset. A pattern here consists of a sequence definition and all of its occurrences in the data. The initial state of the search algorithm is the set of pattern candidates consisting of all uniquely labeled sensor identifiers. The only operators of the search are the *ExtendSequence* operator and the *EvaluatePattern* operator. The *ExtendSequence* operator extends a pattern definition by growing it to include the sensor event that occurs before or after any of the instances of the pattern.

The entire dataset is scanned to create initial patterns of length one. After this first iteration, the whole dataset does not need to be scanned again. Instead, AD extends the patterns discovered in the previous iteration using the *ExtendSequence* operator and will match the extended pattern against the patterns already discovered in the current iteration to see if it is a variation of a previous pattern or is a new pattern. In addition, AD employs an optional pruning heuristic that removes patterns from consideration if the newly-extended child pattern evaluates to a value that is less than the value of its parent pattern.

AD uses a beam search to identify candidate sequence patterns by applying the *ExtendSequence* operator to each pattern that is currently in the open list of candidate patterns. The patterns are stored in a beam-limited open list and are ordered based on their value.

The search terminates upon exhaustion of the search space. Once the search terminates and AD reports the best patterns that were found, the sensor event data can be compressed using the best pattern. The compression procedure replaces all instances of the pattern by single event descriptors, which represent the pattern definition. AD can then be invoked again on the compressed data. This procedure can be repeated a user-specified number of times. Alternatively, the search and compression process can be set to repeat until no new patterns can be found that compress the data. We use the last mode for experiments in this paper.

4.3 Pattern Evaluation

AD's search is guided by the minimum description length (MDL) [47] principle. The evaluation heuristic based on the MDL principle assumes that the best pattern is one that minimizes the description length of the original dataset when it is compressed using the pattern definition. Specifically, each occurrence of a pattern can be replaced by a single event labeled with the pattern identifier. As a result, the description length of a pattern P given the input data D is calculated as $DL(P) + DL(D|P)$, where $DL(P)$ is the description length of the pattern definition and $DL(D|P)$ is the description length of the dataset compressed using the pattern definition. Description length is calculated in general as the number of bits required to minimally encode the dataset. We estimate description length as the number of sensor events that comprise the dataset. As a result, AD seeks a pattern P that maximally compresses the data, or maximizes the value of

$$Compression = \frac{DL(D)}{DL(P) + DL(D|P)}.$$

Because human behavioral patterns rarely occur exactly the same way twice, we employ an edit distance measure to determine if a sensor sequence is an acceptable variation of a current pattern, and thus should be considered as an occurrence of the pattern. This allowance provides a mechanism for finding fewer patterns that abstract over slight variations in how activities are performed.

To determine the fit of a variation to a pattern definition we compute the edit distance using the Damerau-Levenshtein measure [48]. This measure counts the minimum number of operations needed to transform one sequence, x, to be equivalent to another, y. In the case of the Damerau-Levenshtein distance, the allowable transformation operators include change of a symbol (in our case, a sensor event), addition/deletion of a symbol, and transposition of two symbols. AD considers a sensor event sequence to be equivalent to another if the edit distance is less than 0.1 times the size of the longer sequence. The edit distance is computed in time $O(|x| \times |y|)$.

As an example, Figure 3 shows a dataset where the sensor identifiers are represented by varying colors. AD discovers four instances of the pattern P in the data that are sufficiently similar to the pattern definition. The resulting compressed dataset is shown as well as the pattern P' that is found in the new compressed dataset.

4.4 Clustering Patterns

Although the pattern discovery process allows for variations between pattern occurrences, the final set of discovered patterns can still be quite large with a high degree of similarity among the sets of patterns. We want to find even more abstract pattern descriptions to represent the set of pattern activities. The final step of the AD algorithm is therefore to cluster the discovered patterns into this more abstract set.

To cluster the patterns, we employ QT clustering [49] in which patterns are merged based purely on similarity and the number of final clusters does not need to be specified a priori. Similarity in this case is determined based on mutual information of the sensor IDs comprising the cluster patterns and the closeness of the pattern occurrence times. Once the AD pattern discovery and cluster process is complete, we can report the set of discovered activities by expressing the cluster centroids. We can also label occurrences of the patterns in the original dataset or in new streaming data to use for activity recognition.

5 Combining Activity Recognition and Activity Discovery in AD+AR

The use of AD-discovered patterns for activity recognition is shown in Figure 4. Sample sensor data is shown in the figure that AD uses to find frequent patterns. Instances of the frequent patterns (in this case, a pattern with the label "Pat_4") are labeled in the data set in the same way that other sensor events are labeled with predefined activities (in this example, Cook and Eat). Features are extracted for the each sliding-window sequence of 20 sensor events and sent to the AR machine learning model for training. In this case, the activity label for the last event in the window should be Pat_4. After training, the machine learning algorithms is now able to label future sensor events with the corresponding label (in this case the choices would be Cook, Eat or Pat_4).

To consider how AD and AR can work in partnership to improve activity recognition, consider the confusion charts shown in Figures 5 a, b and c. These graphs show how the online SVM classifier performs for the three datasets when only predefined activities are considered (all sensor events not belonging to one of these activities are removed). We include a confusion matrix visualization to indicate where typical misclassifications occur and to highlight how skewed the class distribution is. For each of the datasets, the cooking, hygiene, and (in the case of B3), work activities dominate the sensor events. This does not mean that the most time is spent in these activities, they simply generate the most sensor events. Misclassifications occur among predictably similar activities, such between Sleep and Bed-toilet and between Bathe and Hygiene.

In contrast, Figures 7 a, b and c show the confusion matrices when all of the sensor data is considered. In this case, we do not filter sensor events which do not belong to a predefined class. Instead, we assign them to an “Other” category. The average classification accuracies in this case are 60.55% for B1, 49.28% for B2, and 74.75% for B3. These accuracies are computed only for predefined activities, for which we are particularly interested. The accuracy when the Other class is also considered increases by 15% on average.

As the graphs illustrate, the accuracy performance degrades when the non-labeled data is included in the analysis. There are a couple of reasons for this change in performance. First, the Other class dominates the data, thus many data points that belong to predefined activities are misclassified as Other (this can be seen in the confusion matrix graphs). Second, the Other class itself represents a number of different activities, transitions, and movement patterns. As a result, it is difficult to characterize this complex class and difficult to separate it from the other activity classes.

We hypothesize that in situations such as this where a large number of the data points belong to an unknown or Other class, activity discovery can play a dual role. First, the discovered patterns can help understand the nature of the data itself. Second, discovery can boost activity recognition by separating the large Other class into separate activity classes, one for each discovered activity pattern and a much-reduced Other class.

To validate our hypothesis, we apply the AD+AR discovery algorithm to our three datasets. Our goal is to characterize as much of the Other class as possible, so we repeat the AD discovery-compress process until no more patterns can be found that compress the data. Table 3 summarizes information about discovered patterns and the amount of data that is characterized by these patterns. Figure 6 shows three of the top patterns discovered in the B1 dataset. The first two visualized patterns are transition patterns. In the first case the resident is entering the dining room from the kitchen and next is moving to the bedroom as the resident gets ready to sleep in the evening. The third pattern represents a stretch of time that the resident spends in the secondary bedroom. This pattern has a significant length and number of occurrences but is not a predefined activity, so the pattern occurrences are not labeled in the input dataset.

In the next step, we use AR to learn models for the predefined activities, the discovered activities, and the small Other class. The AD program outputs the sensor data annotated with occurrences of not only the predefined activities but also the discovered activities. This annotated data can then be fed to AR to learn the models. Figures 8 a, b and c show the confusion matrices for the predefined and the other classes without discovered patterns. The accuracies for recognizing the pattern classes are not included for sake of space and to focus on the ability to recognize the activities of primary interest. Table 4 compares the recognition results for predefined activities with an Other class and for predefined activities together with discovered activities and an other class. The improvement due to addition of discovered pattern classes is significant ($p < 0.01$) and is most likely due to the partitioning of the large Other class into subclasses that are more separable from the predefined activities.

6 Conclusions and Future Work

In order to provide robust activity-aware services for real-world applications, researchers need to design techniques to recognize activities in real time from sensor data. This presents a challenge for machine learning algorithms, particularly when not all of the data belongs to a predefined activity class.

In this paper we discussed a method for handling this type of online activity recognition by forming a partnership between activity discovery and activity recognition. In our approach, the AD activity discovery algorithm identifies patterns in sensor data that can partition the undefined class and provide insights on behavior patterns. We demonstrate that treating these discovered patterns as additional classes to learn also improves the accuracy of the AR online activity recognition algorithm.

While this is a useful advancement to the field of activity recognition, there is additional research that can be pursued to enhance the algorithms. Although AD processes the entire data set to find patterns of interest in our experiments, when AD is used in production mode it will only perform discovery on a sample of the data and use the results to boost AR for real-time recognition of new data that is received. As a result, we would like to investigate a streaming version of AD that incrementally refines patterns based on this continual stream of data. We would also like to design methods of identifying commonalities between discoveries in different datasets as well as transferring the discovered activities to new settings to boost activity recognition across multiple environments and residents. By looking for common patterns across multiple settings we may common patterns of interest that provide insight on behavioral characteristics for target population groups.

When we look at the patterns that AD discovers, we notice some similarity between some of the patterns and the predefined activities. However, these occurrences of the predefined activities are not always correctly annotated in the dataset itself (most often occurrences of predefined activities are missed). We hypothesize that the AD+AR approach can be used to identify and correct possible sources of annotation error and thereby improve the quality of the annotated data as well.

Furthermore, we observe ways in which the AR algorithm itself can be improved. By making the window size dependent on the likely activities that are being observed the window size can be dynamic and not reliant upon a fixed value. This is a direction that will be pursued to make real-time activity recognition more adaptive to varying activities and settings.

This study is part of the larger CASAS smart home project. A number of CASAS tools, demos, and datasets can be downloaded from the project web page at http://ailab.wsu.edu/casas to facilitate use, enhancement and comparison of approaches. Tackling the complexities of activity recognition in realistic settings moves this project closer to the goal of providing functional assessment of adults in their everyday settings and providing activity-aware interventions that sustain functional independence. We also believe that examining these challenging issues allows us to consider a wider range of real-world machine learning uses in noisy, sensor-rich applications.

Acknowledgments

We would like to acknowledge support for this project from the National Science Foundation (NSF grant CNS-0852172), the National Institutes of Health (NIBIB grant R01EB009675), and the Life Sciences Discovery Fund.

References

1. Reisburg B, Finkel S, Overall J, Schmidt-Gollas N, Kanowski S, Lehfeld H, Hulla F, Sclan SG, Wilms HU, Heininger K, Hindmarch I, Stemmler M, Poon L, Kluger A, Cooler C, Bergener M, Hugonot-Diener L, robert PH, Erzigkeit H. The Alzheimer's disease activities of daily living international scale (ADL-IS). International Psychogeriatrics. 2001; 13(2):163–181. [PubMed: 11495392]
2. Farias ST, Mungas D, Reed B, Harvey D, Cahn-Weiner D, DeCarli C. MCI is associated with deficits in everyday functioning. Alzheimer Disease and Associated Disorders. 2006; 20:217–223. [PubMed: 17132965]
3. Schmitter-Edgecombe M, Woo E, Greeley D. Characterizing multiple memory deficits and their relation to everyday functioning in individuals with mild cognitive impairment. Neuropsychology. 2009; 23:168–177. [PubMed: 19254090]
4. Wadley V, Okonkwo O, Crowe M, Ross-Meadows LA. Mild cognitive impairment and everyday function: Evidence of reduced speed in performing instrumental activities of daily living. American Journal of Geriatric Psychiatry. 2007; 16:416–424. [PubMed: 18448852]
5. Das, B.; Chen, C.; Seelye, A.; Cook, D. An automated propmting system for smart environments," in. Proceedings of the Internaitonal Conference on Smart Homes and Health Telematics; 2011.
6. Kaushik P, Intille S, Larson K. User-adaptive reminders for home-based medical tasks. a case study. Methods of Information in Medicine. 2008; 47:203–207. [PubMed: 18473085]
7. Cook, DJ.; Das, SK. Smart Environments: Technology, Protocols and Applications. Wiley; 1995.
8. Kim E, Helal A, Cook D. Human activity recognition and pattern discovery. IEEE Pervasive Computing. 2010; 9(1):48–53. [PubMed: 21258659]
9. Maurer, U.; Smailagic, A.; Siewiorek, D.; Deisher, M. Activity recognition and monitoring using multiple sensors on different body positions. Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks; 2006. p. 113-116.
10. Yin J, Yang Q, Pan JJ. Sensor-based abnormal human-activity detection. IEEE Transactions on Knowledge and Data Engineering. 2008; 20(8):1082–1090.
11. Gyorbiro N, Fabian A, Homanyi G. An activity recognition system for mobile phones. Mobile Networks and Applications. 2008; 14:82–91.
12. Kwapisz, JR.; Weiss, GM.; Moore, SA. Activity recognition using cell phone accelerometers. Proceedings of the International workshop on Knowledge Discovery from Sensor Data; 2010. p. 10-18.
13. Gu T, Chen S, Tao X, Lu J. An unsupervised approach to activity recognition and segmentation based on object-use fingerprints. Data and Knowledge Engineering. 2010
14. Palmes P, Pung HK, Gu T, Xue W, Chen S. Object relevance weight pattern mining for activity recognition and segmentation. Pervasive and Mobile Computing. 2010; 6(1):43–57.
15. Philipose M, Fishkin KP, Perkowitz M, Patterson DJ, Fox dieter, Kautz H, Hahnel D. Inferring activities from interactions with objects. IEEE Pervasive Computing. 2004; 3:50–57.
16. Cook D. Learning setting-generalized activity models for smart spaces. IEEE Intelligent Systems. to appear.
17. Logan, B.; Healey, J.; Philipose, M.; Tapia, EM.; Intille, S. A long-term evaluation of sensing modalities for activity recognition. Proceedings of the International Conference on Ubiquitous Computing; 2007.
18. Wang, L.; Gu, T.; Tao, X.; Lu, J. Sensor-based human activity recognition in a multi-user scenario. Proceedings of the European Conference on Ambient Intelligence; 2009. p. 78-87.
19. Yang J, Schilit BN, McDonald DW. Activity recognition for the digital home. Computer. 2008; 41(4):102–104.

20. Brdiczka O, Crowley JL, Reignier P. Learning situation models in a smart home. IEEE Transactions on Systems, Man, and Cybernetics, Part B. 2009; 39(1)

21. Cook DJ, Schmitter-Edgecombe M. Assessing the quality of activities in a smart environment. Methods of Information in Medicine. 2009; 48(5):480–485. [PubMed: 19448886]

22. Tapia EM, Intille SS, Larson K. Activity recognition in the home using simple and ubiquitous sensors. Proceedings of Pervasive. 2004:158–175.

23. van Kasteren, T.; Krose, B. Bayesian activity recognition in residence for elders. Proceedings of the IET International Conference on Intelligent Environments; 2007. p. 209-212.

24. Lombriser, C.; Bharatula, NB.; Roggen, D.; Troster, G. On-body activity recognition in a dynamic sensor network. Proceedings of the International Conference on Body Area Networks; 2007.

25. Liao, IL.; Fox, D.; Kautz, H. Location-based activity recognition using relational Markov networks. Proceedings of the International Joint Conference on Artificial Intelligence; 2005. p. 773-778.

26. Sanchez D, Tentori M, Favela J. Activity recognition for the smart hospital. IEEE Intelligent Systems. 2008; 23(2):50–57.

27. Hu, DH.; Pan, SJ.; Zheng, VW.; Liu, NN.; Yang, Q. Real world activity recognition with multiple goals. Proceedings of the International Conference on Ubiquitous Computing; 2008. p. 30-39.

28. Vail, DL.; Lafferty, JD.; Veloso, MM. Conditional random fields for activity recognition. Proceedings of the International Conference on Autonomous Agens and Multi-agent Systems; 2007. p. 1-8.

29. Fleury, A.; Noury, N.; Vacher, M. Supervised classification of activities of daily living in health smart homes using SVM. Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society; 2009. p. 6099-6102.

30. Gu, T.; Wu, Z.; Tao, X.; Pung, HK.; Lu, J. epSICAR: an emerging patterns based approach to sequential, interleaved and concurrent activity recognition. Proceedings of the IEEE International Conference on Pervasive Computing and Communications; 2009. p. 1-9.

31. Chiang, Y-T.; Hsu, K-C.; Lu, C-H.; Fu, L-C. Interaction models for multiple-resident activity recognition in a smart home. Proceedings of the International Conference on Intelligent Robots and Systems; 2010. p. 3753-3758.

32. Phua, C.; Sim, K.; Biswas, J. Multiple people activity recognition using simple sensors. Proceedings of the International Conference on Pervasive and Embedded Computing and Communication Systems; 2011. p. 224-231.

33. Chang CC, Lin CJ. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology. 2011; 2(27):1–27.

34. Agrawal, R.; Srikant, R. Mining sequential patterns. Proceedings of the International Conference on Data Engineering; 1995. p. 3-14.

35. Barger T, Brown D, Alwan M. Health-status monitoring through analysis of behavioral patterns. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans. 2005; 35(1): 22–27.

36. Pei J, Han J, Wang W. Constraint-based sequential pattern mining: The pattern-growth methods. Journal of Intelligent Information Systems. 2007; 28(2):133–160.

37. Asier A, Augusto J, Cook D. Discovering frequent user-environment interactions in intelligent environments. Personal and Ubiquitous Computing. to appear.

38. Heierman, EO.; Cook, DJ. Improving home automation by discovering regularly occurring device usage patterns. Proceedings of the IEEE International Conference on Data Mining; 2003. p. 537-540.

39. Ruotsalainen, A.; Ala-Kleemola, T. Gais: A method for detecting discontinuous sequential patterns from imperfect data. Proceedings of the International Conference on Data Mining; 2007. p. 530-534.

40. Pei, J.; Han, J.; Asl, MB.; Pinto, H.; Chen, Q.; Dayal, U.; Hsu, MC. Prefixspan: Mining sequential patterns efficiently by prefix projected pattern growth. Proceedings of International Conference on Data Engineering; 2001. p. 215-226.

41. Zaki, MJ.; Lesh, N.; Ogihara, M. Planmine: Sequence mining for plan failures. Proceedings of the International Conference on Knowledge Discovery and Data Mining; 1998. p. 369-373.

42. Chen YI, Chen SS, Hsu PY. Mining hybrid sequential patterns and sequential rules. Information Systems. 2002; 27(5):345–362.

43. Rashidi P, Cook D, Holder L, Schmitter-Edgecombe M. Discovering activities to recognize and track in a smart environment. IEEE Transactions on Knowledge and Data Engineering. 2011; 23(4):527–539. [PubMed: 21617742]

44. Huynh, T.; Fritz, M.; Schiele, B. Discovery of activity patterns using topic models. Proceedings of the International Conference on Ubiquitous Computing; 2008. p. 10-19.

45. Cook D, Holder L. Graph-based data mining. IEEE Intelligent Systems. 2000; 15(2):32–41.

46. Yoshida K, Motoda H, Indurkhya N. Graph-based induction as a unified learning framework. Journal of Applied Intelligence. 1994; 4:297–328.

47. Rissanen, J. Stochastic Complexity in Statistical Inquiry. World Scientific Publishing Company; 1989.

48. Levenshtein VI. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady. 1966; 10(8):707–710.

49. Heyer LJ, Kruglyak S, Yooseph S. Exploring expression data: Identification and analysis of coexpressed genes. Genome Research. 1999; 9(11):1106–1115. [PubMed: 10568750]

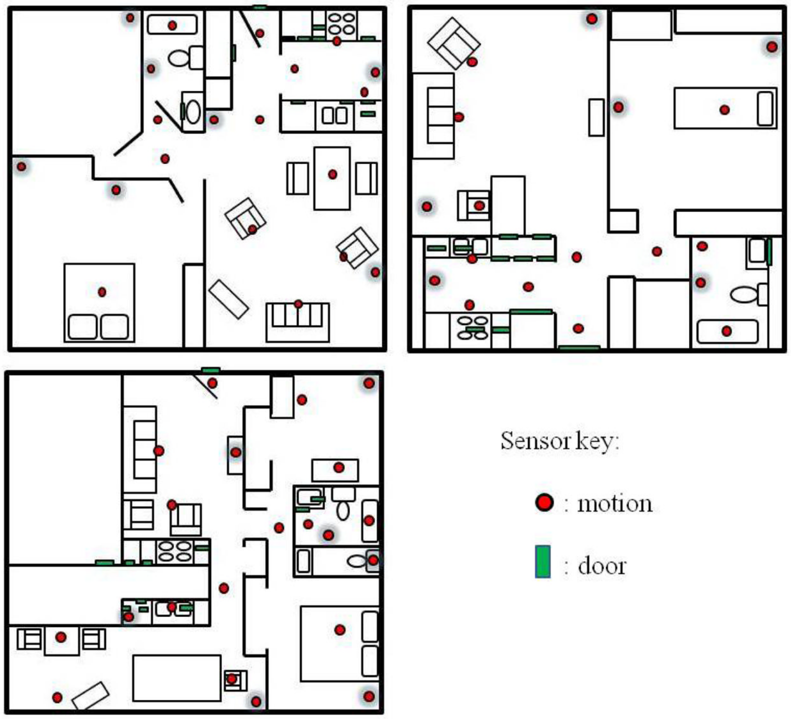


Fig. 1.
Floorplans for the B1, B2, and B3 testbeds.

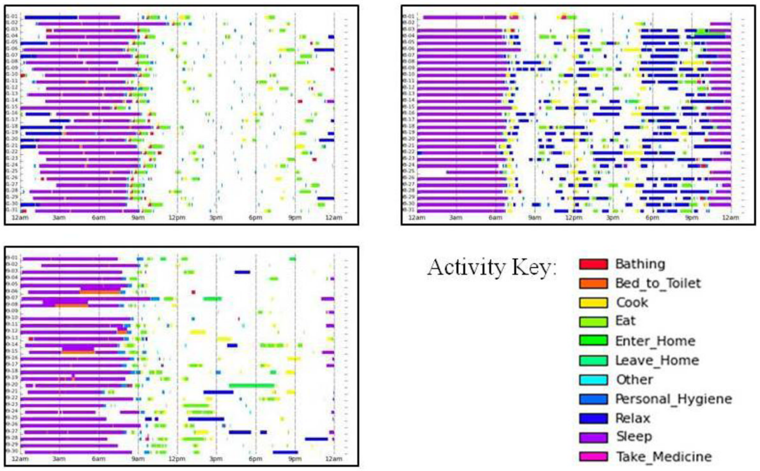


Fig. 2.

Plot of activity occurrences for the three testbeds. The x axis represents time of day starting at midnight, and the y axis represents a specific day.

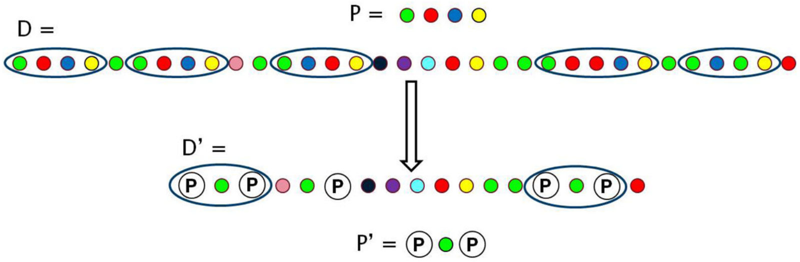
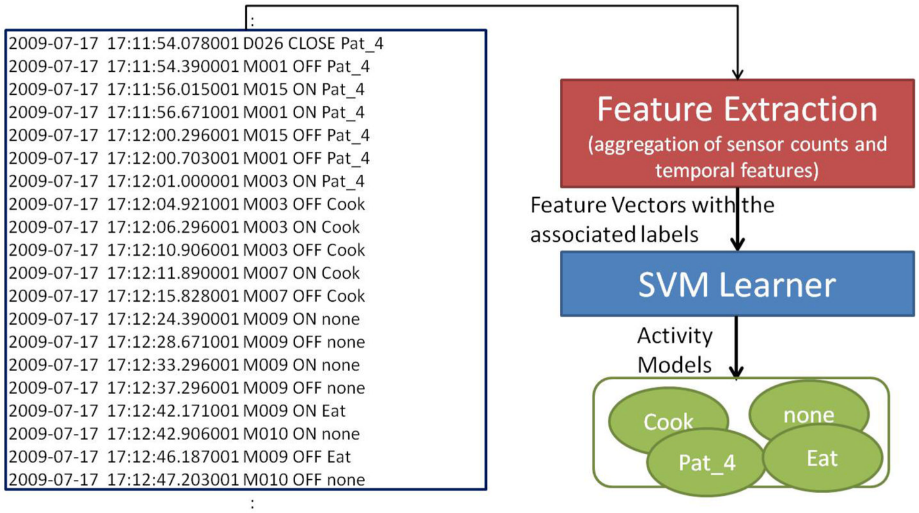


Fig. 3.
Example of the AD discovery algorithm. A sequence pattern (*P*) is identified and used to compress the dataset. A new best pattern (pattern *P*′) is found in the next iteration of the algorithm.

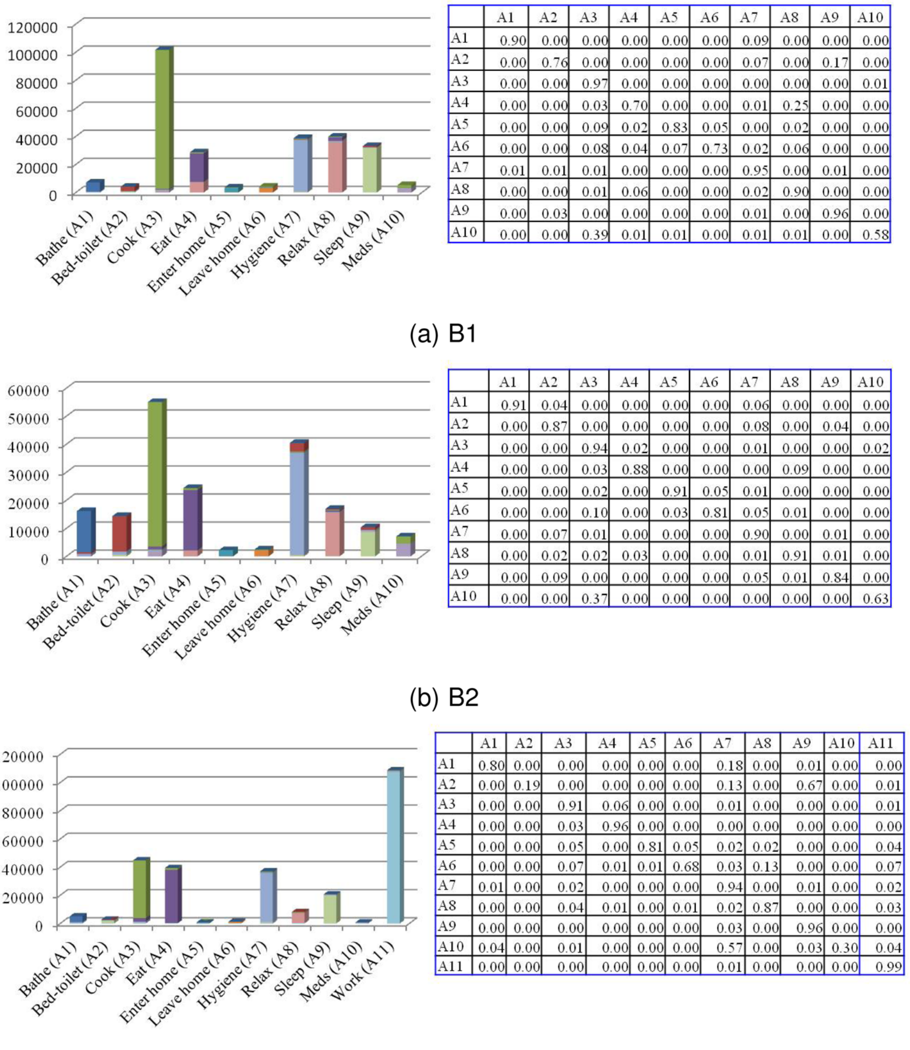


```
:
2009-07-17  17:11:54.078001 D026 CLOSE Pat_4
2009-07-17  17:11:54.390001 M001 OFF Pat_4
2009-07-17  17:11:56.015001 M015 ON Pat_4
2009-07-17  17:11:56.671001 M001 ON Pat_4
2009-07-17  17:12:00.296001 M015 OFF Pat_4
2009-07-17  17:12:00.703001 M001 OFF Pat_4
2009-07-17  17:12:01.000001 M003 ON Pat_4
2009-07-17  17:12:04.921001 M003 OFF Cook
2009-07-17  17:12:06.296001 M003 ON Cook
2009-07-17  17:12:10.906001 M003 OFF Cook
2009-07-17  17:12:11.890001 M007 ON Cook
2009-07-17  17:12:15.828001 M007 OFF Cook
2009-07-17  17:12:24.390001 M009 ON none
2009-07-17  17:12:28.671001 M009 OFF none
2009-07-17  17:12:33.296001 M009 ON none
2009-07-17  17:12:37.296001 M009 OFF none
2009-07-17  17:12:42.171001 M009 ON Eat
2009-07-17  17:12:42.906001 M010 ON none
2009-07-17  17:12:46.187001 M009 OFF Eat
2009-07-17  17:12:47.203001 M010 OFF none
:
```

Feature Extraction
(aggregation of sensor counts and temporal features)
Feature Vectors with the associated labels
SVM Learner
Activity Models
Cook
none
Pat_4
Eat

Fig. 4.

Flowchart for the AD+AR algorithm.



	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	0.90	0.00	0.00	0.00	0.00	0.00	0.09	0.00	0.00	0.00
A2	0.00	0.76	0.00	0.00	0.00	0.00	0.07	0.00	0.17	0.00
A3	0.00	0.00	0.97	0.00	0.00	0.00	0.00	0.00	0.00	0.01
A4	0.00	0.00	0.03	0.70	0.00	0.00	0.01	0.25	0.00	0.00
A5	0.00	0.00	0.09	0.02	0.83	0.05	0.00	0.02	0.00	0.00
A6	0.00	0.00	0.08	0.04	0.07	0.73	0.02	0.06	0.00	0.00
A7	0.01	0.01	0.01	0.00	0.00	0.00	0.95	0.00	0.01	0.00
A8	0.00	0.00	0.01	0.06	0.00	0.00	0.02	0.90	0.00	0.00
A9	0.00	0.03	0.00	0.00	0.00	0.00	0.01	0.00	0.96	0.00
A10	0.00	0.00	0.39	0.01	0.01	0.00	0.01	0.01	0.00	0.58

(a) B1

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	0.91	0.04	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.00
A2	0.00	0.87	0.00	0.00	0.00	0.00	0.08	0.00	0.04	0.00
A3	0.00	0.00	0.94	0.02	0.00	0.00	0.01	0.00	0.00	0.02
A4	0.00	0.00	0.03	0.88	0.00	0.00	0.00	0.09	0.00	0.00
A5	0.00	0.00	0.02	0.00	0.91	0.05	0.01	0.00	0.00	0.00
A6	0.00	0.00	0.10	0.00	0.03	0.81	0.05	0.01	0.00	0.00
A7	0.00	0.07	0.01	0.00	0.00	0.00	0.90	0.00	0.01	0.00
A8	0.00	0.02	0.02	0.03	0.00	0.00	0.01	0.91	0.01	0.00
A9	0.00	0.09	0.00	0.00	0.00	0.00	0.05	0.01	0.84	0.00
A10	0.00	0.00	0.37	0.00	0.00	0.00	0.00	0.00	0.00	0.63

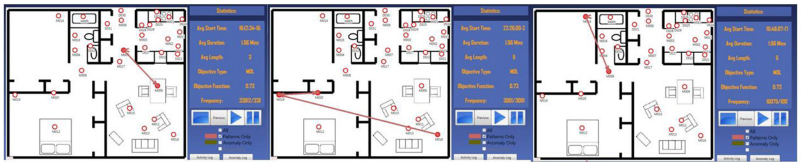
(b) B2

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11
A1	0.80	0.00	0.00	0.00	0.00	0.00	0.18	0.00	0.01	0.00	0.00
A2	0.00	0.19	0.00	0.00	0.00	0.00	0.13	0.00	0.67	0.00	0.01
A3	0.00	0.00	0.91	0.06	0.00	0.00	0.01	0.00	0.00	0.00	0.01
A4	0.00	0.00	0.03	0.96	0.00	0.00	0.00	0.00	0.00	0.00	0.00
A5	0.00	0.00	0.05	0.00	0.81	0.05	0.02	0.02	0.00	0.00	0.04
A6	0.00	0.00	0.07	0.01	0.01	0.68	0.03	0.13	0.00	0.00	0.07
A7	0.01	0.00	0.02	0.00	0.00	0.00	0.94	0.00	0.01	0.00	0.02
A8	0.00	0.00	0.04	0.01	0.00	0.01	0.02	0.87	0.00	0.00	0.03
A9	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.96	0.00	0.00
A10	0.04	0.00	0.01	0.00	0.00	0.00	0.57	0.00	0.03	0.30	0.04
A11	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.99

(c) B3

Fig. 5.

Confusion charts for the three datasets, shown by raw number of data points classified for each label (left) and percentage of data points classified for each label (right).



Statistics:

Avg Start Time: 16:12:34-16

Avg Duration: 1.56 Mins

Avg Length: 3

Objective Type: MDL

Objective Function: 0.73

Frequency: 33612/336

□ All
☑ Patterns Only
□ Anomaly Only

Statistics:

Avg Start Time: 22:26:00-2

Avg Duration: 1.56 Mins

Avg Length: 5

Objective Type: MDL

Objective Function: 0.73

Frequency: 3001/3001

□ All
☑ Patterns Only
□ Anomaly Only

Statistics:

Avg Start Time: 10:48:07-17

Avg Duration: 1.56 Mins

Avg Length: 5

Objective Type: MDL

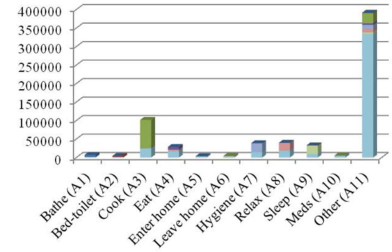
Objective Function: 0.73

Frequency: 10075/100

□ All
☑ Patterns Only
□ Anomaly Only

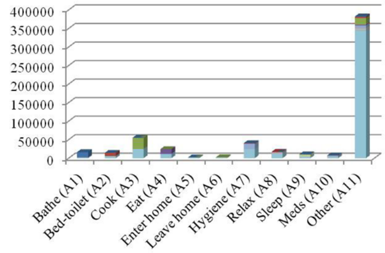
Fig. 6.

Three top patterns discovered in B1 dataset.



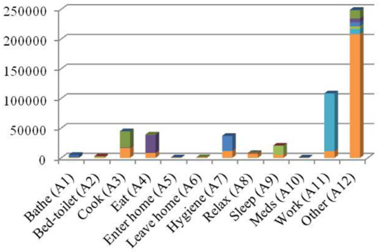
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11
A1	0.77	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.18
A2	0.00	0.74	0.00	0.00	0.00	0.00	0.06	0.00	0.14	0.00	0.06
A3	0.00	0.00	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25
A4	0.00	0.00	0.01	0.26	0.00	0.00	0.00	0.13	0.00	0.00	0.60
A5	0.00	0.00	0.02	0.00	0.14	0.00	0.00	0.00	0.00	0.00	0.84
A6	0.00	0.00	0.01	0.00	0.01	0.12	0.00	0.00	0.00	0.00	0.86
A7	0.01	0.01	0.01	0.00	0.00	0.00	0.62	0.00	0.01	0.00	0.35
A8	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.52	0.00	0.00	0.47
A9	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.68	0.00	0.29
A10	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.88
A11	0.00	0.00	0.07	0.01	0.00	0.00	0.03	0.02	0.01	0.00	0.85

(a) B1



	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11
A1	0.87	0.01	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.09
A2	0.00	0.59	0.00	0.00	0.00	0.00	0.03	0.00	0.01	0.00	0.37
A3	0.00	000	0.55	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.45
A4	0.00	0.00	0.00	0.53	0.00	0.00	0.00	0.00	0.00	0.00	0.47
A5	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.00	0.00	0.00	0.88
A6	0.00	0.00	0.01	0.00	0.00	0.08	0.00	0.00	0.00	0.00	0.91
A7	0.00	0.03	0.00	0.00	0.00	0.00	0.37	0.00	0.01	0.00	0.58
A8	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.21	0.00	0.00	0.77
A9	0.00	0.04	0.00	0.00	0.00	0.00	0.02	0.00	0.59	0.00	0.35
A10	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.35	0.65
A11	0.00	0.01	0.05	0.01	0.00	0.00	0.02	0.00	0.01	0.00	0.90

(b) B2

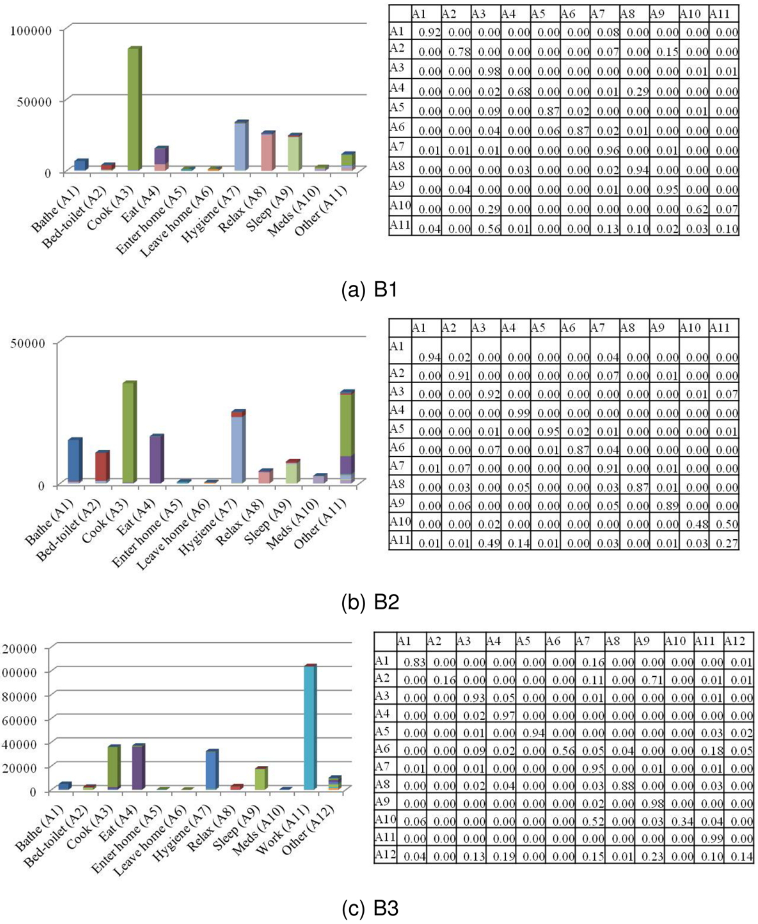


	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
A1	0.74	0.00	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.00	0.00	0.14
A2	0.00	0.10	0.00	0.00	0.00	0.00	0.06	0.00	0.51	0.00	0.01	0.33
A3	0.00	0.00	0.61	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.37
A4	0.00	0.00	0.01	0.78	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.21
A5	0.00	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.01	0.73
A6	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.01	0.93
A7	0.00	0.00	0.00	0.00	0.00	0.00	0.67	0.00	0.00	0.00	0.01	0.31
A8	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.18	0.00	0.00	0.00	0.81
A9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.00	0.00	0.25
A10	0.03	0.00	0.00	0.00	0.00	0.00	0.18	0.00	0.00	0.13	0.00	0.66
A11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.89	0.10
A12	0.00	0.00	0.06	0.03	0.00	0.00	0.03	0.00	0.02	0.00	0.03	0.84

(c) B3

Fig. 7.

Confusion charts for the three datasets with Other class, shown by raw number of data points classified for each label (left) and percentage of data points classified for each label (right).



	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11
A1	0.92	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.00	0.00	0.00
A2	0.00	0.78	0.00	0.00	0.00	0.00	0.07	0.00	0.15	0.00	0.00
A3	0.00	0.00	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01
A4	0.00	0.00	0.02	0.68	0.00	0.00	0.01	0.29	0.00	0.00	0.00
A5	0.00	0.00	0.09	0.00	0.87	0.02	0.00	0.00	0.00	0.01	0.00
A6	0.00	0.00	0.04	0.00	0.06	0.87	0.02	0.01	0.00	0.00	0.00
A7	0.01	0.01	0.01	0.00	0.00	0.00	0.96	0.00	0.01	0.00	0.00
A8	0.00	0.00	0.00	0.03	0.00	0.00	0.02	0.94	0.00	0.00	0.00
A9	0.00	0.04	0.00	0.00	0.00	0.00	0.01	0.00	0.95	0.00	0.00
A10	0.00	0.00	0.29	0.00	0.00	0.00	0.00	0.00	0.00	0.62	0.07
A11	0.04	0.00	0.56	0.01	0.00	0.00	0.13	0.10	0.02	0.03	0.10

(a) B1

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11
A1	0.94	0.02	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00
A2	0.00	0.91	0.00	0.00	0.00	0.00	0.07	0.00	0.01	0.00	0.00
A3	0.00	0.00	0.92	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.07
A4	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00
A5	0.00	0.00	0.01	0.00	0.95	0.02	0.01	0.00	0.00	0.00	0.01
A6	0.00	0.00	0.07	0.00	0.01	0.87	0.04	0.00	0.00	0.00	0.00
A7	0.01	0.07	0.00	0.00	0.00	0.00	0.91	0.00	0.01	0.00	0.00
A8	0.00	0.03	0.00	0.05	0.00	0.00	0.03	0.87	0.01	0.00	0.00
A9	0.00	0.06	0.00	0.00	0.00	0.00	0.05	0.00	0.89	0.00	0.00
A10	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.48	0.50
A11	0.01	0.01	0.49	0.14	0.01	0.00	0.03	0.00	0.01	0.03	0.27

(b) B2

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
A1	0.83	0.00	0.00	0.00	0.00	0.00	0.16	0.00	0.00	0.00	0.00	0.01
A2	0.00	0.16	0.00	0.00	0.00	0.00	0.11	0.00	0.71	0.00	0.01	0.01
A3	0.00	0.00	0.93	0.05	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00
A4	0.00	0.00	0.02	0.97	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
A5	0.00	0.00	0.01	0.00	0.94	0.00	0.00	0.00	0.00	0.00	0.03	0.02
A6	0.00	0.00	0.09	0.02	0.00	0.56	0.05	0.04	0.00	0.00	0.18	0.05
A7	0.01	0.00	0.01	0.00	0.00	0.00	0.95	0.00	0.01	0.00	0.01	0.00
A8	0.00	0.00	0.02	0.04	0.00	0.00	0.03	0.88	0.00	0.00	0.03	0.00
A9	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.98	0.00	0.00	0.00
A10	0.06	0.00	0.00	0.00	0.00	0.00	0.52	0.00	0.03	0.34	0.04	0.00
A11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.00
A12	0.04	0.00	0.13	0.19	0.00	0.00	0.15	0.01	0.23	0.00	0.10	0.14

(c) B3

Fig. 8.

Confusion charts for the three datasets with discovered patterns and Other class, shown by number of data points classified for each label (left) and percentage of data points classified for each label (right).

TABLE 1

Characteristics of the three datasets used for this study.

Dataset	**B1**	**B2**	**B3**
#Sensors	32	32	32
#Days Monitored	202	234	177
#Sensor Events	658,811	572,255	518,759
Activity Occurrences	5,714	4,320	3,361

TABLE 2

Characteristics of the three datasets used for this study.

Dataset	B1	B2	B3	Average
NB	92.91%	90.74%	88.81%	90.82%
HMM	92.07%	89.61%	90.87%	90.85%
CRF	85.09%	82.66%	90.36%	86.04%
SVM	90.95%	89.35%	94.26%	91.52%

TABLE 3

Statistics of patterns found for B1, B2, and B3.

Dataset	**B1**	**B2**	**B3**
%Data in Other Class (before compression)	59.45%	66.83%	48.04%
#Discovered patterns	67	45	52
#Pattern clusters	19	18	16
%Data in Other Class (after compression)	4.00%	10.25%	7.05%

TABLE 4

Recognition accuracy for predefined activities with and without activity discovery.

Dataset	B1	B2	B3
No patterns	60.55%	49.28%	74.75%
With patterns	71.08%	59.76%	84.89%