Learning Activity Models for Multiple Agents in a Smart Space

Aaron Crandall and Diane J. Cook

1 Introduction

With the introduction of more complex intelligent environment systems, the possibilities for customizing system behavior have increased dramatically. Significant headway has been made in tracking individuals through spaces using wireless devices [1, 18, 26] and in recognizing activities within the space based on video data (see chapter by Brubaker et al. and [6, 8, 23]), motion sensor data [9, 25], wearable sensors [13] or other sources of information [14, 15, 22]. However, much of the theory and most of the algorithms are designed to handle one individual in the space at a time. Resident tracking, activity recognition, event prediction, and behavior automation becomes significantly more difficult for multi-agent situations, when there are multiple residents in the environment.

The goal of this research project is to model and automate resident activity in multiple-resident intelligent environments. There are simplifications that would ease the complexity of this task. For example, we could ask residents to wear devices that enable tracking them through the space [7, 17, 26]. This particular solution is impractical for situations in which individuals do not want to wear the device, forget to wear the device, or enter and leave the environment frequently. Similarly, video cameras are valuable tools for understanding resident behaviors, even in group settings [2, 10]. However, surveys with target populations have revealed that many individuals are adverse to embedding cameras in their personal environments [8]. As a result, our aim is to identify the individuals and their activities in an intelligent environment using passive sensors.

Aaron S. Crandall
School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA, USA e-mail: acrandal@eecs.wsu.edu

Diane J. Cook
School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA, USA e-mail: cook@eecs.wsu.edu

To achieve this overall goal, our first step is to design an algorithm that maps sensor events to the resident that is responsible for triggering a sensor event. This information will allow our algorithms to learn profiles of resident behaviors, identify the individuals currently in the environment, monitor their well-being, and automate their interactions with the environment. Some earlier approaches allow for multiple residents in a single automated setting [4]. In addition, a number of algorithms have been successful designed for activity recognition, albeit in a single-resident setting [12].

To date, however, the focus has often been on looking at global behaviors and preferences with the goal of keeping a group of residents satisfied [21]. In contrast, our research is focused on identifying an individual and logging their preferences and behaviors in the context of the multi-resident spaces.

The solutions used in this work revolve around using very simple passive sensors, such as motion, contact, door sensors, appliance interaction and light switches to give a picture of what is transpiring in the space. These information sources offer the benefits of being fixed, unobtrusive and robust devices. Examples of the motion detectors and light switches we use in our testbed are shown in Figure 3.

Smart homes are often targeted towards recognizing and assisting with the Activities of Daily Living (ADLs) that the medical community uses to categorize levels of healthy behavior in the home [5, 16, 20]. The ability of smart homes to help disabled and elderly individuals to continue to operate in the familiar and safe environment is one of the greatest reasons for their continued development. So far, most smart home research has focused on monitoring and assisting a single individual in a single space. Since homes often have more than a single occupant, building solutions for handling multiple individuals is vital. Dealing with multiple residents has rarely been the central focus of research so far, as there have been numerous other challenges to overcome before the technology can effectively handle multiple residents in a single space.

Since smart home research has the ultimate goal of being deployable in real-world environments, seeking solutions that are as robust as possible is always a factor in the systems we engineered. With that in mind, building an entirely passive solution gives the advantage of keeping the technology separate from the residents while they go about performing their daily routines. This lets the smart home feel as "normal" as possible to the residents and their guests. By reducing the profile of the new devices as much as possible, people should be less affected by the technology that surrounds them.

In this chapter we present a solution to part of the problem described above. Specifically, apply a supervised machine learning algorithm to the task of mapping sensor events to the resident responsible for the event. The solution proposed in this work offers the advantage of using previous behavioral data collected from the set of known residents without requiring significant additional actions to be performed by the residents. This historical behavior is used to train the learning algorithm for use in future real-time classification of the individuals and can be updated over time as new data arrives.

Here we present the results of using a naive Bayesian classifier to learn resident identities based on observed sensor data. Because this machine learning algorithm is probabilistic, likelihood values are generated for each resident that can be used to appropriately modify the behavior of the intelligent environment. Because the algorithm is efficient and robust, we hypothesize that it will be able to accurately handle the problem of learning resident identities and be usable in a real-time intelligent environment. We validate our hypothesis using data collected in a real smart workplace environment with volunteer participants.

2 Testbeds

The smart home testbed environments at Washington State University consist of a lab space on campus and a town home off campus. These testbeds are part of WSU's CASAS smart environments project [19]. For our study, we used the lab space on campus, as there are multiple faculty, staff, and students who regularly enter the space and a number of different kinds of activity take place throughout the rooms. The space is designed to capture temporal and spatial information via motion, door, temperature and light control sensors.

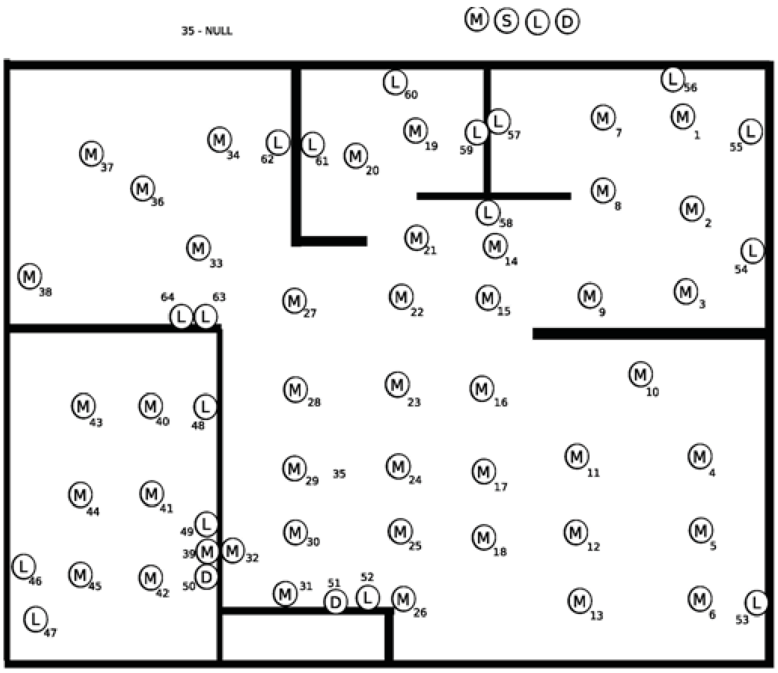

Fig. 1 Layout of motion (M), door (D), and light switch (S) sensors in the testbed environment.

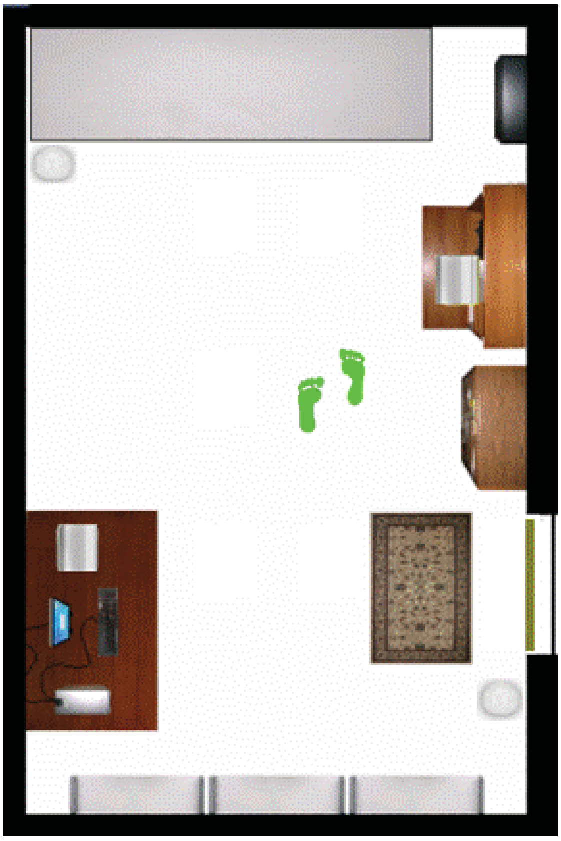

Fig. 2 2D view of inner office furniture layout.

For this project we focus on events collected from motion sensors and resident interaction with lighting devices. The testbed layout is shown in Figures 1 and 2. Throughout this space, motion detectors are placed on the ceilings and pointed straight down, as shown in Figure 3. Their lenses are occluded to a smaller rectangular window giving them roughly a 3’ x 3’ coverage area of the corresponding floor space. By placing them roughly every four feet, they overlap (between a few inches, up to a foot) and allow tracking of an individual moving across the space. The motion sensor units are able to sense when a motion as small as reaching from the keyboard to a mouse. With this level of sensitivity, sensors around workspaces trip even when people sit quietly in a private space to work at a computer.

To provide control and sensing over the lighting, Insteon switches are used to control all of the ceiling and desk lights in the room. These switches communicate with a computer and all interactions with them are logged. See Figure 3 for images of both the motion and light switch sensors. The entire lab space, including the portion shown in Figure 2, has two doors with simple magnetic open/closed sensors

Fig. 3 CASAS motion sensor and Insteon light switch controller.

affixed to them. These door sensors record door openings and closings via the same bus as the motion detectors.

By being able to log any major movement throughout the space, as well as device interactions, this system captures basic temporal and spatial behaviors that can be used to identify individuals based on behavior. Residents in this lab have unique work spaces, as well as unique time frames within which they operate. The tools used in this project are designed to exploit both the spatial and temporal differences between individuals to accurate classify a given individual. These are features of most kinds of living spaces and can be used by software algorithms to accurately identify the current resident.

3 Data Representation

The data gathered by CASAS for this study is represented by a quintuple:

1. Date
2. Time
3. Sensor Serial Number
4. Event Message
5. Annotated Class (Resident ID)

The first four fields are generated automatically by the CASAS data collection infrastructure. The annotated class field is the target field for this problem and represents the resident ID, to which the sensor event can be mapped.

Training data was gathered during several weeks in the lab space by asking three individuals working in the lab to log their presence. These participants registered their presence in the lab by pushing a unique button on a pinpad when they entered and again when they left the space. During post processing, the database was filtered to only use sensor events during the time windows when there was a single resident in the space. The corresponding data for the given time frame was then annotated and supplied as training data to our machine learning algorithm. The total time frame for data collection was three weeks, and over 6000 unique events were captured and annotated as training data. A sample of the data as quintuples is shown in Table 1.

Table 1 Sample of data used for classifier training.

Date	*Time*	*Serial*	*Message*	*ID*
2007-12-21	16:41:41	07.70.eb:1	ON	Abe
2007-12-21	16:44:36	07.70.eb:1	OFF	Abe
2007-12-24	08:13:50	E9.63.a7:5	ON	John
2007-12-24	14:31:30	E9.63.a7:5	OFF	John

One aspect of the data that presents a particular challenge for supervised machine learning algorithms is the time value. In order to insightfully reason about the time at which events occur, we created more complex features designed to capture the differences in behavior between individuals. Primarily, these strategies revolved around using the data and time information to give the classifier additional information in the form of "feature types", as shown in Table 2. The times that different people work, especially in a research lab, are very helpful in discriminating the likely resident that is currently in the space. In our lab, one of the three participants worked late at night on several occasions, while another of the participants was the only one to ever arrive before 10am. By incorporating temporal information into the features, this kind of behavior can improve the accuracy of the classifier. Given an automatic training system, picking the best feature type(s) to use can be based on a combination of the resulting accuracy and false positive rates.

Table 2 Feature types used to represent time values for classifier training.

Feature Type	*Example*
Simple	`07.70.eb:1#ON`
Hour of Day	`07.70.eb:1#ON#6`
Day of Week	`07.70.eb:1#ON#FRI`
Part of Day	`07.70.eb:1#ON#MORNING`
Part of Week	`07.70.eb:1#ON#WEEKDAY`

4 Classifier

We selected a naive Bayes classifier for our learning problem. These kinds of classifiers have been used with great effect in other smart home research projects [24] with success. Applying the same kind of tool to individual identification from the same kind of data set is a logical extension. In this case, a simple naive Bayes classifier was trained, where the features were built from the event information. The target class is the individual with whom the event is associated. This required it be distilled to only a single feature paired to a given class. The class is set by the annotation, but the feature chosen can be built from a number of the fields.

For the simplest interpretation, only the serial number coupled with an event message was used, as in Row 1 of Table 2. This simple feature set provides a good baseline to compare with more complex parsings. The more complex parsings, such as "Part-of-Week" (i.e., WEEKDAY or WEEKEND) capture more information about the given behavior, and can give the classifier more information for correct future classifications. Depending on the facets of the data set, different kinds of feature types can give the classifier better or worse results.

The data set was randomly split into training and testing sets, with 10% of each class set aside for testing. The classifier was trained on the 90% and run against the testing set. Each class was given an accuracy rate and a false positive rate. This process was repeated for each of our feature types for comparison of accuracy and false positive rates.

Training the classifier followed a simple naive Bayes algorithm, as shown in Equation 1.

$$Likelihood(Person_n|Event_i) = P(Person_n) * P(Event_i|Person_n) \quad (1)$$

In this case, $Event_i$ is defined by what kind of feature type is being used (See Table 2 for the ones used in this study). So, the likelihood that a particular person, n, generated sensor event i can be calculated as the product of the probability that person n generates a sensor event (independent of a particular event) and the probability that person n generates event i.

The different feature choices available (i.e., Simple vs. Hour of Day) split the data up in different ways. Each way captures the behaviors or the residents with varying degrees of accuracy, depending on the feature types chosen and the behavior of the individuals in the data set. The purely statistical nature of a naive Bayes classifier has the benefit of being fast for use in prediction engines, but lacks the ability to handle context in the event stream that could be advantageous in discerning different behaviors.

The naive Bayes classifier is a bag-of-sensor-data learning algorithm, which neither explicitly represents nor reasons about the temporal and spatial relationships between the data points. In contrast, a Markov model encompasses this type of information and, therefore, may do a better job of performing the classification task. A Markov Model (MM) is a statistical model of a dynamic system. A MM models the system using a finite set of states, each of which is associated with a multidimensional probability distribution over a set of parameters. The system is assumed to be a Markov process, so the current state depends on a finite history of previous states (in our case, the current state depends only on the previous state). Transitions between states are governed by transition probabilities. For any given state a set of observations can be generated according to the associated probability distribution. Because our goal is to identify the individual that caused the most recent in a sequence of sensor events, we generate one Markov model for each smart environment resident whose profile we are learning. We use the training data to learn the transition probabilities between states for the corresponding activity model and to learn probability distributions for the feature values of each state in the model.

A Markov model processes a sequence of sensor events. We feed the model with a varying-length sequence of sensor readings that lead up to and include the most recent reading. To label the most recent reading with the corresponding resident that caused the sensor event, we compute $Person_n$ as $argmax_{n \in N} P(n|e_{1..t}) = P(e_{1..t}|n)P(n)$. $P(Person_n)$ is estimated as before, while $P(e_{1..t}|n)$ is the result of computing the sum, over all states, S, in model n, of the likelihood of being in each state after processing the sequence of sensor events $e_{1..t}$. The likelihood of being in state $s \in S$ is updated after each sensor event e_j is processed using the formula found in Equation 2. The probability is updated based on the probability of transitioning from any previous state to the current state (the first term of the summation) and the probability of being in the previous state given the sensor event sequence that led up to event e_j.

$$P(S_j|e_{1..j}) = P(e_j|S_j)\Sigma s_{j-1} P(S_j|s_{j-1})P(s_{j-1}|e_{1..j-1}) \quad (2)$$

We learn a separate Markov model for each resident in the smart environment. To classify a sensor event, the model that best supports the sensor event sequence is identified and the corresponding resident is output as the label for the most recent sensor event.

5 Results

Figure 4 shows the classification accuracy of our naive Bayes classifier for the three residents we tested in our lab space. In order to keep actual participant names anonymous, we label the three residents John, Abe, and Charlie. In Figure 4 we graph not only the classification accuracy for each target value, but also the false positive rate.

Note that the classification accuracy is quite high for the John values, but so is the false positive rate. This is because our John participant was responsible for most (roughly 62%) of the sensor events in the training data. As a result, the apriori probability that any sensor event should be mapped to John is quite high and the naive Bayes classifier incorrectly attributes Abe and Charlie events to John as well. On the other hand, while Charlie has a much lower correct classification rate, he also has

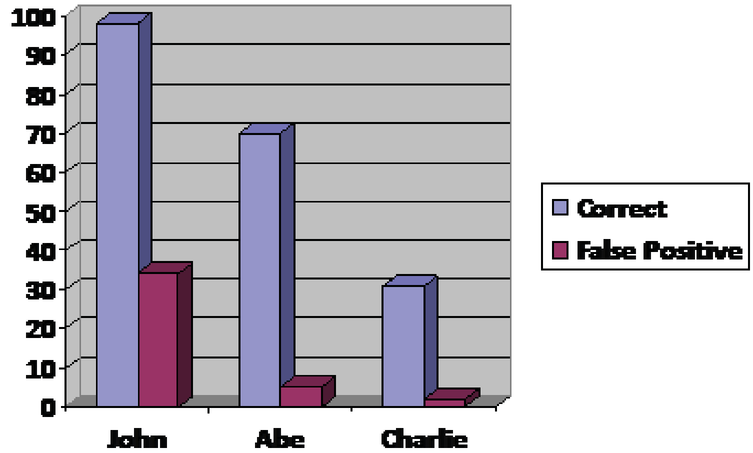

Fig. 4 Classification accuracy using all sensor events.

a lower false positive rate. If the intelligent environment can take likelihood values into account, this information about false positives can be leveraged accordingly.

In order to address these classification errors, we added more descriptive features to our data set. In particular, we added the date and time of each sensor event, as shown in Table 2. The classifier can now use time of day or day of week information to differentiate between the behaviors of the various individuals. For example, John always arrived early in the day, while Abe was often in the space late into the evening. Finding the correct features to use for this kind of capturing of the behavior can be done by balancing the overall correct classification rate and false positive rate against one another.

The choice of feature descriptors to use is quite important and has a dramatic effect on the classification accuracy results. Looking at the accuracy rate as affected by the feature type chosen, Figure 5, it shows that using hour-of-day increases the average identification significantly. Additionally, by using hour-of-day, the false positive rate drops dramatically, as shown in Figure 6. When the right features are selected from the data set, the classifier is able to make better overall classifications.

To demonstrate the effects of picking the best time based identification features for our machine learning problem on an individual's accuracy, refer to Figure 7. The first column represents the simplest feature set (Table 2, row 1), but comparing it against using the hour-of-day (Table 2, row 2) shows readily that if the classifier is given this extra information John's accuracy percentage barely moves, while his false positive rate drastically drops, as shown in Figure 8. The false positive rate actually drops from 34% to 9%, which is a marked improvement.

Use of the other time based features results in some improvements to John's classification, but none of the others is others as useful as adding the hour-of-day feature. As an example that has accuracy improvements, but has tradeoffs, Charlie's

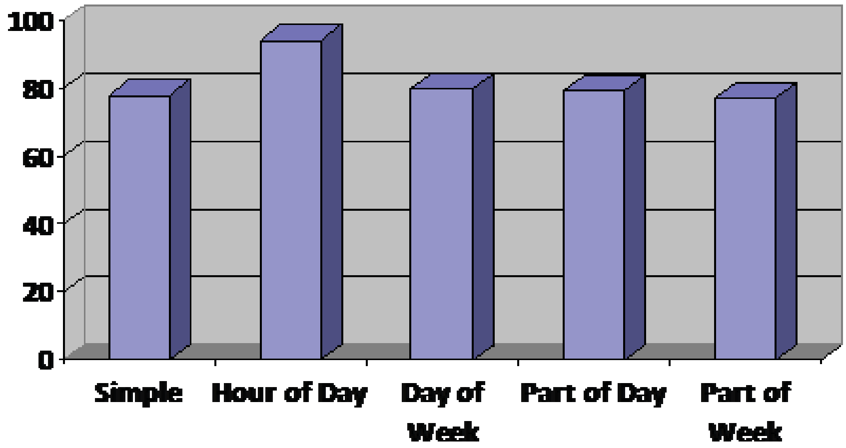

Fig. 5 Average classification accuracy for alternative feature types.

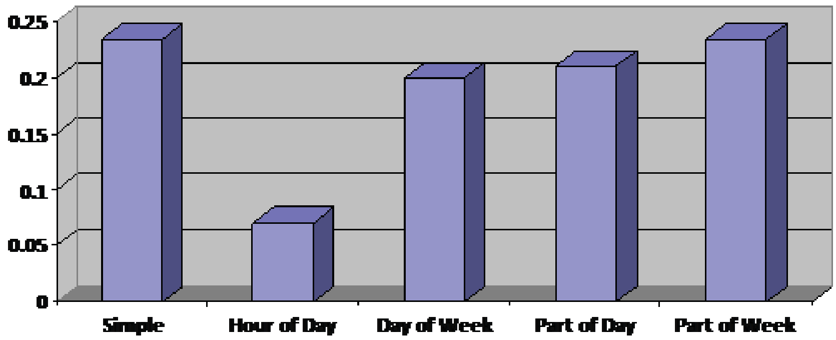

Fig. 6 Average false positive rates for alternative feature types.

behavior responds differently to the choice of feature type. To demonstrate the improvements in accuracy rate, refer to Figure 9. Charlie's initial 31% accuracy with simple features was shown to jump to 87% by again using the hour-of-day feature type.

This is again likely due to the times of day when Charlie's activities do not overlap as much with Abe or John. The cost in this example is that Charlie's rate of false positives goes up from 3% to 6%, as shown in Figure 10. This type of tradeoff needs to be taken into account when performing feature selection for any classification task. Choosing the best feature type to pick involves balancing classification accuracy against the false positive rate. A visual representation of this balancing task is shown in Figure 11. By choosing the time-of-day feature, the benefits to the accuracy rate will probably outweigh the increase in false positive rate. In this case,

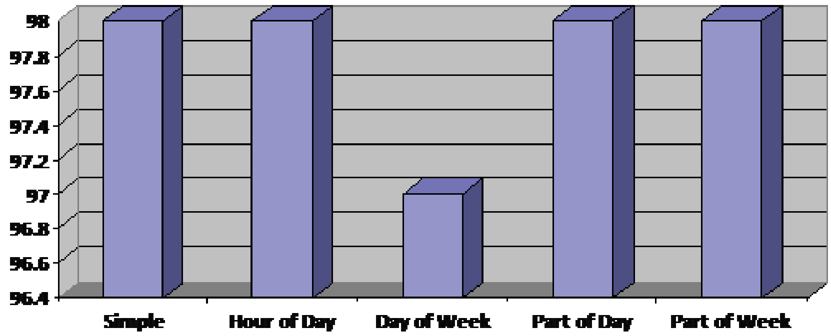

Fig. 7 Classification accuracy for John class across feature types.

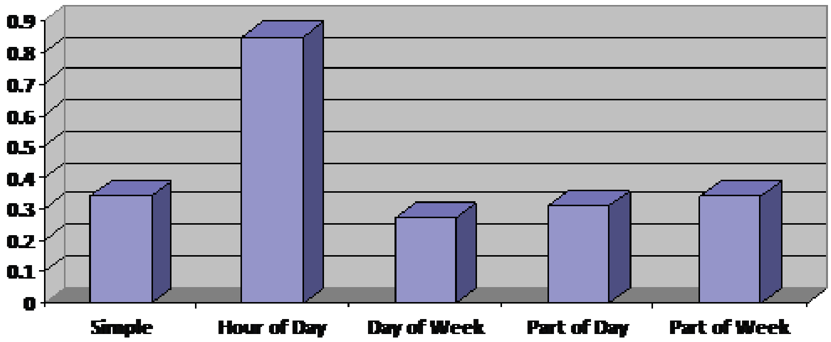

Fig. 8 False positive rates for John class across feature types.

a 2.5x increase in accuracy balances against a 2x increase in false positives. Unless the final smart environment use of the classification is highly dependent on the certainty of the predictions, it should be a simple algorithm to determine which feature type is most advantageous.

For this data set, the other features including day of week, part-of-day and part-of-week have little improvement over the simple feature strategy. With a correctness of over 93% and a false positive rate below 7%, a prediction engine relying on this classifier can have a high degree of confidence that it is correctly choosing the proper preferences for a given individual.

5.1 Time Delta Enhanced Classification

Adding more features to our data set did improve the resident classification accuracy. However, the results were still not as good as we anticipated. We hypothesize

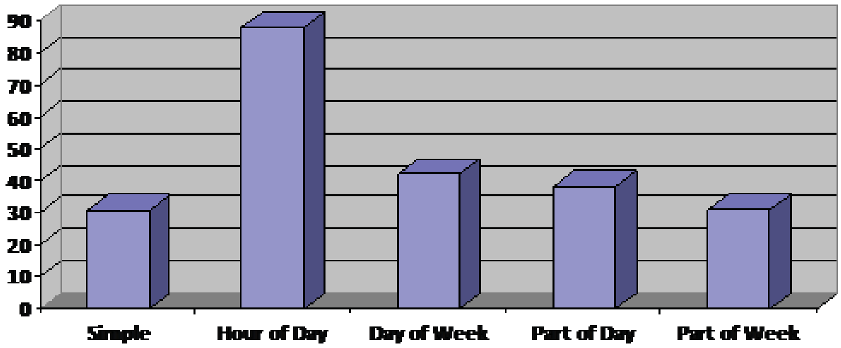

Fig. 9 Classification accuracy for Charlie class across feature types.

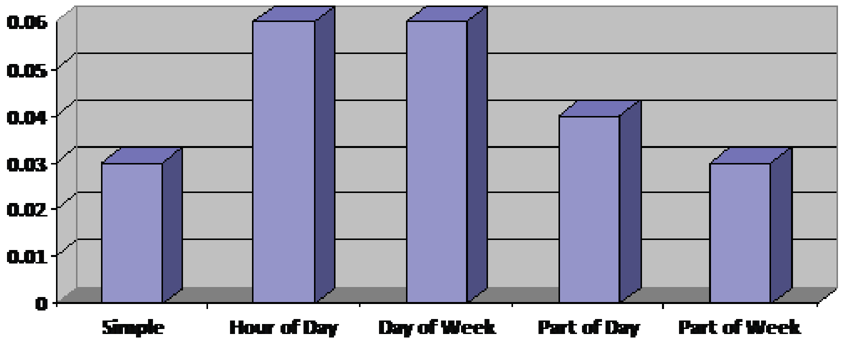

Fig. 10 False positive rates for Charlie class across feature types.

that one reason for the remaining inaccuracies is the type of sensor events we are classifying. Many motion sensor events occur when individuals are moving through the space to get to a destination, and do not differentiate well between residents in the space. On the other hand, when a resident is in a single location for a significant amount of time, that location is a type of destination for the resident. They are likely performing an activity of interest in that location, and as a result the corresponding motion sensor data should be used for resident classification.

To validate our hypothesis, the data set was culled of all extra sensor events where the same sensor generated multiple readings in a row and only the first event in the series was kept. The multiple readings were likely due to small movements occurring repeatedly within the one small area of the lab. Replacing the set of readings with one representative motion sensor event allowed the sensor event to represent the entire activity taking place at that location.

With this reduced set of events, the time deltas, or time elapsed between the remaining events, were calculated. The chart shown in Figure 12 gives a count of how long an individual spent at any one motion sensor location before moving to a

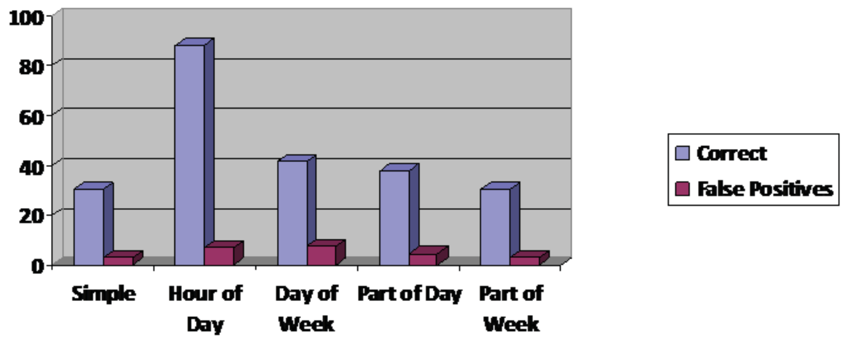

Fig. 11 Overall classification rates for Charlie class across feature types.

new location. The average time spent on any sensor was 35 seconds, with a standard deviation of 10 seconds. With a graph of this shape, we have evidence supporting the initial hypothesis of being able to garner additional information for training by focusing on locations with significant duration.

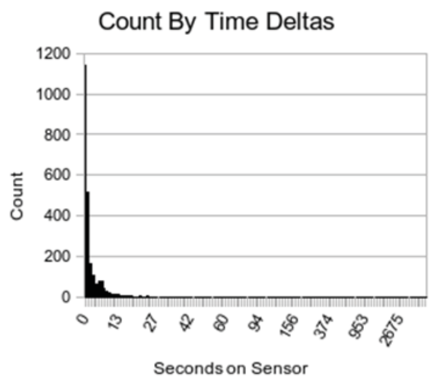

Fig. 12 The number of motion sensor events that spanned a given number of seconds.

Next, we removed from our data set any motion sensor events whose durations, or time elapsed between events, fell below two standard deviations from the mean, leaving the longest deltas. With an even more reduced set in hand, the data splitting, training and testing were all done the same way as before with the full data set.

The resulting classifier only used a handful of the available sensors throughout the living space, but the accuracy and false positive rates improved dramatically. This is attributed to the fact that motion sensors in shared spaces or walkways will mostly have very small time deltas associated with them. Since these sensors are also the ones with the most false positive rates in the full set classifier, removing these sensor events will improve the overall performance of the classifier. Note that with this filtered-data approach, sensor events with short durations will not be assigned a

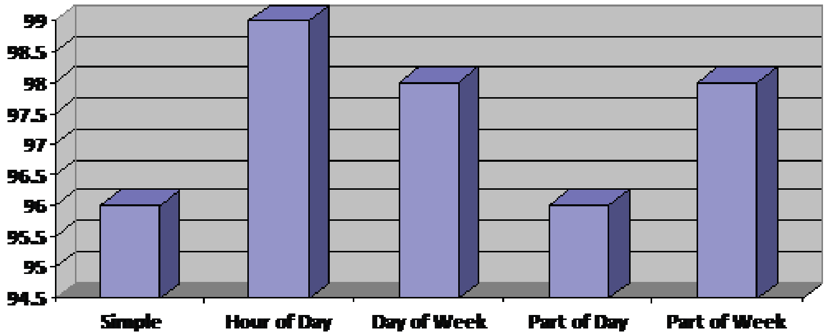

Fig. 13 Delta-filtered classification accuracy results.

mapping to a specific resident. However, by combining this tool with one that tracks residents through the space [7], only a handful of sensor events need to be classified as long as they have a high accuracy.

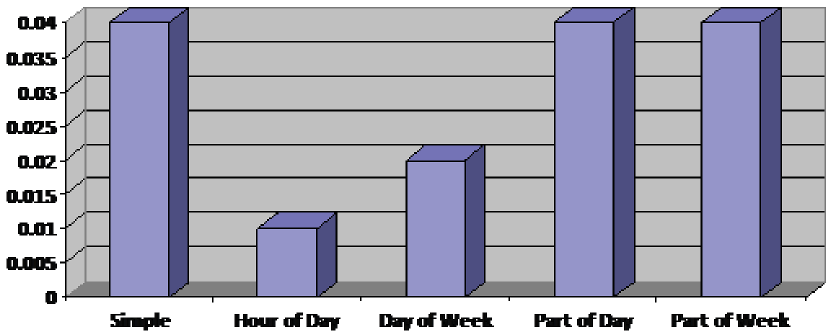

Fig. 14 Delta-filtered classification false positive results.

This new classifier saw correct classification rates over 98% with false positives as low as 1%. Again, there was some difference in performance with different feature choices, as shown in Figures 13 and 14. Once again, the hour-of-day feature performed the best, as it seems to give the naive Bayes classifier information that could be used to differentiate between resident behaviors within this data set.

5.2 Markov Model Classification Accuracy

In our Markov model approach to labeling sensor events, we need to determine the number of sensor events that we will include in the sequence input to each Markov

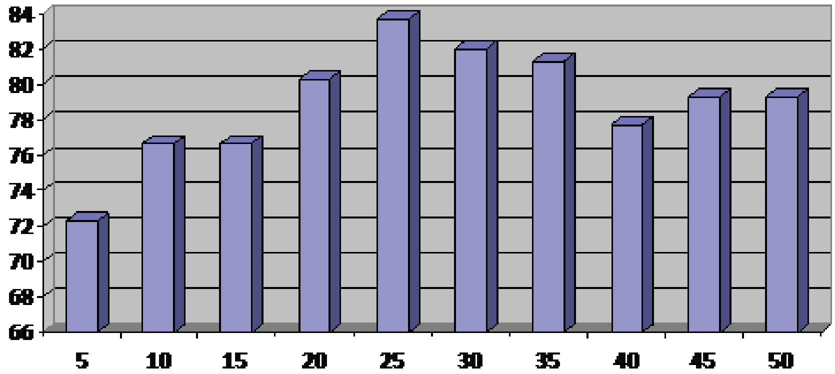

Fig. 15 Markov model classification accuracy for all three residents as a function of trace size.

model. There is no magic formula that specifies the ideal size for such an event trace. As a result, we computed the accuracy of the Markov model classifier for a variety of event trace sizes. The results are shown in Figure 15. As the figure shows, the classification performance peaks at a trace size of 25. None of the accuracies outperform the naive Bayes classifier for this task. We note that finding the ideal trace size needs to be automated for future applications of the Markov model. One possible approach is to build a model on a sample of the data and empirically determine the trace size that yields the best result for a given classification task.

The results from these experiments indicate that resident behavior histories can be insightful mechanisms for identifying the resident that triggered a sensor event, even when multiple residents are present simultaneously in an environment. We did find that some sensor events (e.g., those with long durations) are easier to characterize than others, because they are more closely associated with an activity that the resident may be performing. We also observed that the more complex Markov model did not outperform the naive Bayes classifier. This indicates that viewing the classification task as a bag-of-sensor-events model is effective for identifying residents who are currently in the smart environment and who are triggering sensor events.

Once they are able to learn profiles of resident activities, even in multi-resident settings, the smart environment will still face the challenge of trying to meet the needs of all the residents. Because these needs and preferences may diverge or even compete, there is not necessarily a straightforward solution to such decisions. Lin and Fu [11] learn a model of users' preferences that represents the relationships among users and the dependency between services and sensor observations. By explicitly learning relationships between these entities, their algorithm can automatically select the service that is appropriate for a given resident and context. Because their approach infers the best service at the right time and place, the comfort of residents can be maintained, even in a multi-resident setting.

The competition for resources can be observed even at a software component level. In the work of Colley and Stacey [3], the need for software agents to communicate is recognized. Communication in a complex setting such as a smart environment can become quite costly, when the amount of gathered data and the number of provided services increases. In some circumstances it is advantageous for software agents to form clusters, or associate themselves with other agents with whom they will communicate often. Colley and Stacey investigate alternative voting schemes to design optimal clusters for this type of agent organization that meet an optimal number of needs and preference requests.

While the previous approaches have demonstrated success in negotiating needs of multiple agents in smart environments, there are many situations in which not all of the stated needs or preferences can be met. In response, the smart environment could prioritize the needs or the residents themselves. An alternative approach, put forward by Roy, et al. [21], focuses not on meeting all of the needs of all of the agents, but instead to strike a balance between multiple diverse preferences. The goal of this approach is to achieve a Nash equilibrium state so that predictions and actions achieve sufficient accuracy in spite of possible correlations or conflicts. As demonstrated in this work, the result can be effective for goals such as minimizing energy usage in a smart environment.

6 Conclusions

In this chapter, we investigated a multi-agent issue that pervades smart environment research, that of attributing sensor events to individuals in a multi-resident setting. To address the challenge, we introduced an approach that used a naive Bayes classifier to map sensor events to residents. We felt that a classifier of this type would be able to accurately and quickly differentiate between residents in a smart home environment. Using a real-world testbed with real-world activity, a classifier was built and tested for accuracy.

The results shown in this work are encouraging. With simple, raw smart home sensor data the classifier was showing an average accuracy over 90% for some feature selections. After applying some filtration to the data set to exaggerate the behavior of the residents, accuracy rates over 95% and false positive rates under 2% were possible. Choosing the best time-based features can strongly influence the performance of any temporally-dependent environment, and this is no exception. Whether the final application needs a very high level of certainty for one or more of the residents or can trade that certainty off for higher accuracy across all individuals is up to the needs of the final smart home application. Fortunately, developing an algorithmic way of determining the proper features to use is easily done.

This method of assigning sensor events to residents also add valuable information to tracking systems that identify people based on an observed series of events by taking into account the likelihoods at every stage of a person's behavior. Continued work combining simple tools like these are leading to very strong identification strategies. To continue to grow the capabilities of these kinds of classifiers, a number of things can help. Additional data with more individuals will show how robust of a solution this is. Differentiating between two people with very similar schedules might be very difficult for this kind of tool. Comparing this tool as a baseline solution with Hidden Markov or Bayesian Network based solutions will allow the continued research to show how much contextual information assists with the classification of individuals.

A next step for this research is to couple the classifier with a larger preference and decision engine. Adding this tool to a passive tracking solution will give significantly more information to any individual's history for future learning and prediction systems that are deployed in the CASAS testbed. Comparing it to a system without this kind of identification process, or one based on device tracking, will be a significant step for smart home research.

Acknowledgements This work is supported in part by NSF grant IIS-0121297.

References

[1] P. Bahl and V. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *Procedings of IEEE Infocom*, pages 775–784, 2000.

[2] O. Brdiczka, P. Reignier, and J. Crowley. Detecting individual activities from video in a smart home. In *Proceedings of the International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 363–370, 2007.

[3] M. Colley and P. Stacey. Agent association group negotiation within intelligent environments. In *Proceedings of the International Conference on Intelligent Environments*, 2008.

[4] D. Cook and S. K. Das. How smart are our environments? an updated look at the state of the art. *Journal of Pervasive and Mobile Computing*, 3(2):53–73, 2007.

[5] M. Diehl, M. Marsiske, A. Horgas, A. Rosenberg, J. Saczynski, and S. Willis. The revised observed activities of daily living: A performance-based assessment of everyday problem solving in older adults. *Journal of Applied Gerontology*, 24(3):211–230, 2005.

[6] W. Feng, J. Walpole, W. Feng, , and C. Pu. Moving towards massively scalable video-based sensor networks. In *Proceedings of the Workshop on New Visions for Large-Scale Networks: Research and Applications*, 2001.

[7] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 32(8):57–66, 2001.

[8] S. S. Intille. Designing a home of the future. *IEEE Pervasive Computing*, 1:80–86, 2002.

[9] V. Jakkula, A. Crandall, and D. J. Cook. Knowledge discovery in entity based smart environment resident data using temporal relations based data mining.

In *Proceedings of the ICDM Workshop on Spatial and Spatio-Temporal Data Mining*, 2007.

[10] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easyliving. In *Proceedings of the Third IEEE International Workshop on Visual Surveillance*, pages 3–10, 2000.

[11] Z. Lin and L. Fu. Multi-user preference model and service provision in a smart home environment. In *Proceedings of the IEEE International Conference on Automation Science and Engineering*, pages 759–764, 2007.

[12] C. Lu, Y. Ho, and L. Fu. Creating robust activity maps using wireless sensor network in a smart home. In *Proceedings of the Third Annual Conference on Automation Science and Engineering*, 2007.

[13] U. Maurer, A. Smailagic, D. Siewiorek, and M. Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*, 2006.

[14] S. Moncrieff. Multi-modal emotive computing in a smart house environment. *Journal of Pervasive and Mobile Computing, special issue on Design and Use of Smart Environments (to appear)*, 2007.

[15] R. J. Orr and G. D. Abowd. The smart floor: A mechanism for natural user identification and tracking. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, The Hague, Netherlands, 2000.

[16] M. B. Patterson and J. L. Mack. The cleveland scale for activities of daily living (csadl): Its reliability and validity. *Journal of Clinical Gerontology*, 7:15–28, 2001.

[17] M. Philipose, K. P. Fishkin, M. Perkowitz, D. J. Patterson, D. Hahnel, D. Fox, and H. Kautz. Inferring activities from interactions with objects. *IEEE Pervasive Computing*, 3(4):50–57, 2004.

[18] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location support system. In *Proceedings of the International Conference on Mobile Computing and Networking*, pages pages 32–43, 2000.

[19] P. Rashidi and D. Cook. Adapting to resident preferences in smart environments. In *Proceedings of the AAAI Workshop on Advances in Preference Handling*, 2008.

[20] B. Reisberg, S. Finkel, J. Overall, N. Schmidt-Gollas, S. Kanowski, H. Lehfeld, F. Hulla, S. G. Sclan, H.-U. Wilms, K. Heininger, I. Hindmarch, M. Stemmler, L. Poon, A. Kluger, C. Cooler, M. Bergener, L. Hugonot-Diener, P. H. Robert, and H. Erzigkeit. The alzheimer's disease activities of daily living international scale. *International Psychogeriatrics*, 13:163–181, 2001.

[21] N. Roy, A. Roy, K. Basu, and S. K. Das. A cooperative learning framework for mobility-aware resource management in multi-inhabitant smart homes. In *Proceedings of the IEEE Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous)*, pages 393–403, 2005.

[22] D. Sanchez, M. Tentori, and J. Favela. Activity recognition for the smart hospital. *IEEE Intelligent Systems*, 23(2):50–57, 2008.

[23] L. Snidaro, C. Micheloni, and C. Chivedale. Video security for ambient intelligence. *IEEE Transactions on Systems, Man and Cybernetics, Part A:Systems and Humans*, 35:133–144, 2005.

[24] T. van Kasteren and B. Krose. Bayesian activity recognition in residence for elders. In *Proceedings of the International Conference on Intelligent Environments*, 2008.

[25] C. Wren and E. Munguia Tapia. Toward scalable activity recognition for sensor networks. In *Proceedings of the Workshop on Location and Context-Awareness*, 2006.

[26] J. Yin, Q. Yang, and D. Shen. Activity recognition via user-trace segmentation. *ACM Transactions on Sensor Networks*, 2008.