Attributing Events to Individuals in Multi-Inhabitant Environments

Aaron S. Crandall and Diane Cook

School of Electrical Engineering and Computer Science
Washington State University, 99164, USA.
acrandal@wsu.edu, cook@eecs.wsu.edu

Keywords: multiple inhabitants, activity recognition, user modeling

Abstract

Intelligent environment research has resulted in many useful tools such as activity recognition, prediction, and automation. However, most of these techniques have been applied in the context of a single resident. A current looming issue for intelligent environment systems is performing these same techniques when multiple residents are present in the environment. In this paper we investigate the problem of attributing sensor events to individuals in a multi-resident intelligent environment. Specifically, we use a naïve Bayesian classifier to identify the resident responsible for a unique sensor event. We present results of experimental validation in a real intelligent workplace testbed and discuss the unique issues that arise in addressing this challenging problem.

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][12][17] and in recognizing activities within the space base on video data [4][6][14], motion sensor data [8][16], or other sources of information [10][11]. However, much of the theory and most of the algorithms are designed to handle one individual in the space at a time. Passive tracking, activity recognition, event prediction, and behavior automation becomes significantly more difficult 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 [5][17]. 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, capturing resident behavior with video cameras aids in understanding resident behavior even in group settings [3]. However, surveys with target populations have revealed that many individuals are adverse to embedding cameras in their personal environments [6]. As a result, our aim is to identify the individuals and their activities in an intelligent environment using passive sensors.

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 previous works have focused on passive multi-resident systems [2], and give some indication of techniques that have succeeded on real-world data sets for activity recognition [9].

To date, the focus has often been on looking at global behaviors and preferences with the goal of keeping a group of inhabitants satisfied [13]. 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 (ADL's) that the medical community uses to categorize levels of healthy behavior in the home. 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 inhabitants has rarely been the central focus of research so far, as their 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 technol-

ogy separate from the inhabitants 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 effected by the technology that surrounds them.
In this paper 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 naïve 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 Data Gathering Environment

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 WSUs CASAS smart environments project. 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. For this project we focus on events collected from motion sensors and resident interaction with lighting devices. Part of the testbed layout for both sensors and furniture is shown in Figures 1 and 2. The rest of the space is very similar with desks, tables and cubicles being the predominate features.
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'x3' 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,

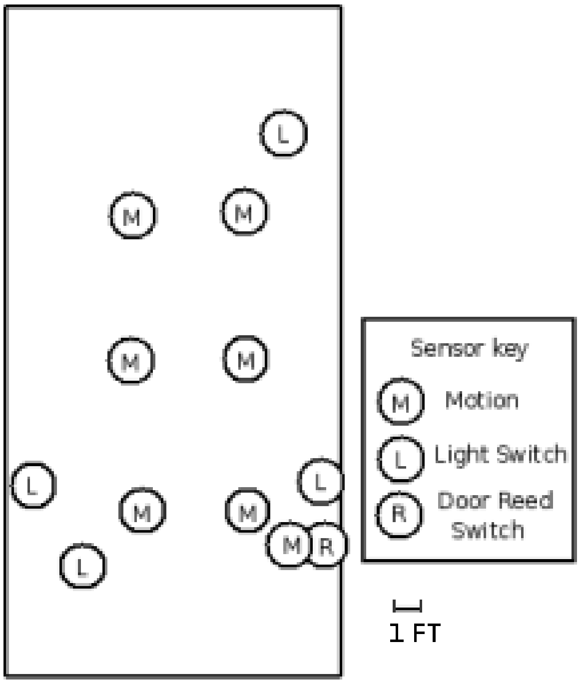

Figure 1: Inner office space sensor layout.

Insteon^{TM}brand 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 affixed to them. These record door openings and closings via the same bus as the motion detectors.
By being able to log any major movement through out 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 inhabitant.

3 Data Representation

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

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

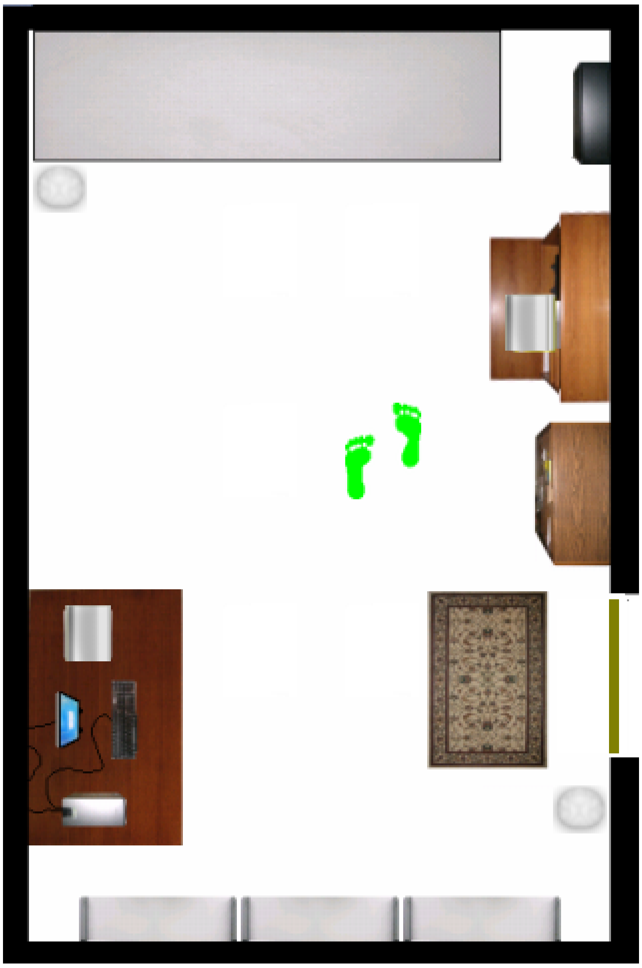

Figure 2: 2D view of inner office furniture.

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

Table 1: Example of data used for classifier training.

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 individuals working in the lab to log their presence by pushing a unique button on a pinpad when they entered and 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. For an example of the resulting quintuples, see Table 1.

Figure 3: CASAS sensors: motion detector and Insteon light switch.

Building more complex parsings of the data was done with a number of strategies that were 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 student 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 a number of times, 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.

4 Classifier

The classifier used for this research is a naïve Bayes. These kinds of classifiers have been used with great effect in other smart home research projects [15] with great 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 naïve Bayes classifier was trained, where the features were built from the event information, with the given class as the individual to whom the event is associated with. 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 event message was used, see Table 2, row 1. This simple feature set provides a good baseline to compare more complex parsings with. The more complex parsings, such as ”Part-of-Week” (ie WEEKDAY or WEEKEND) capture more information about the given behavior, and can give the classifier more information for correct fu-

#	*Feature Type*	*Example*
1	Simple	07.70.eb:1#ON
2	Hour of Day	07.70.eb:1#ON#16
3	Day of Week	07.70.eb:1#ON#FRI
4	Part of Week	07.70.eb:1#ON#WEEKDAY
5	Part of Day	07.70.eb:1#ON#AFTERNOON

Table 2: Feature types used for classifier training

ture 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 naïve Bayes algorithm, as shown in Equation 1.

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

In this case, the $Event_i$ is defined by what kind of feature type is being used (See Table 2 for the ones used in this study). So, the likelyhood that a given event belongs to a person is the probability of that person in the total data set, times the probability of the event given that person.

The different feature choices available (ie Simple vs Hour of Day, etc.) 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 naïve 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.

5 Results

Figure 4 shows the classification accuracy of our naïve Bayesian 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 naïve Bayesian 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 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 this classification errors, we added more

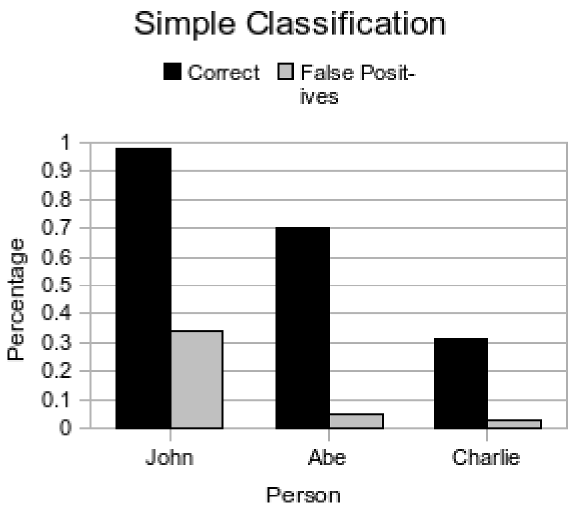

Figure 4: Simple classification with all events.

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 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 effected 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.

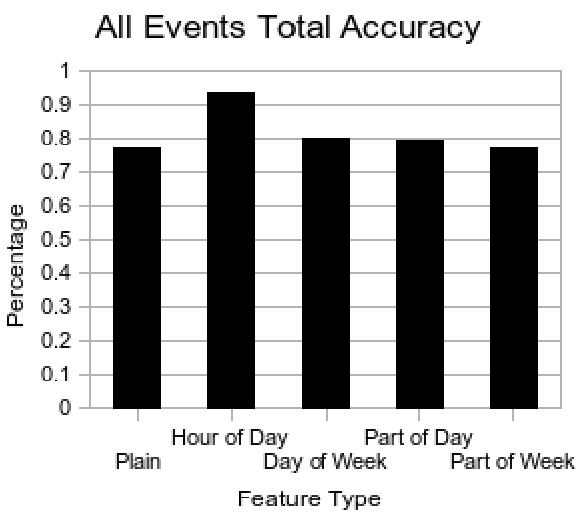

Figure 5: Average accuracy rates by feature type.

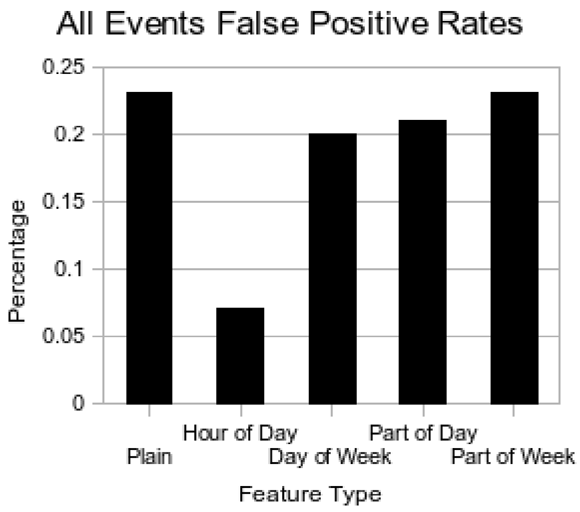

Figure 6: Average false positive rates by feature type.

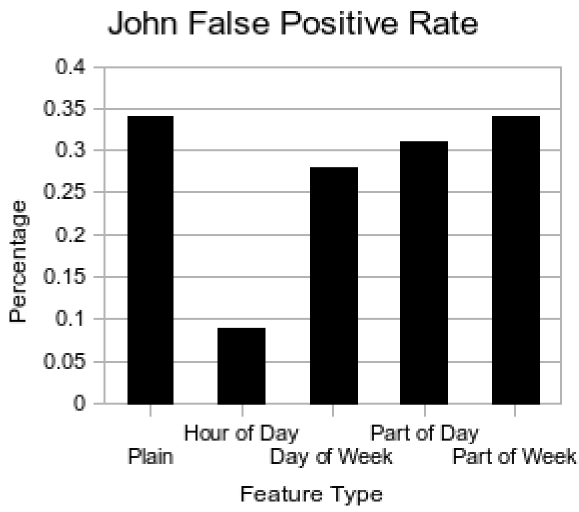

Figure 8: John's rate of false positives across feature types.

To demonstrate the effects of picking the best time based identification features for our 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.

Figure 7: John's rate of correct classification across feature types.

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 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.

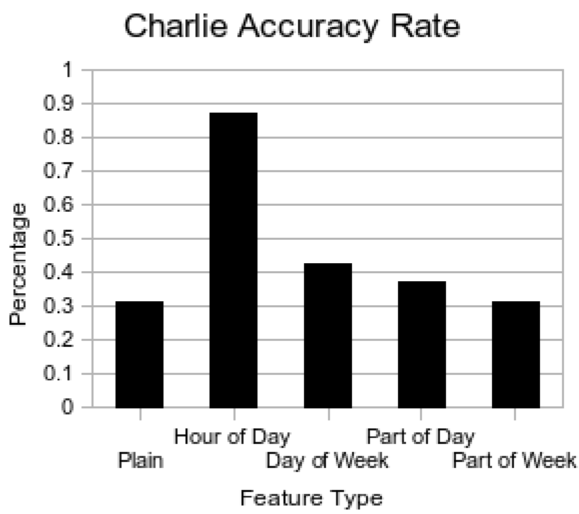

Figure 9: Charlie's rate of correct classification across feature types.

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 kind of trade off needs to be taken into account by any system of deciding which features to use for the current classifier.

Choosing the best feature type to pick means balancing the accuracy against the false positive rate. A visual way of showing this kind of balancing is shown in Figure 11. By choosing time-of-day the benefits to the accuracy rate will probably outweigh the increase in false positive rate. In this case, a 2.5x increase in accuracy balances against a 2x increase in false positives. Unless the final application is highly dependent on the certainty of the predictions, it should be a simple algorithm to determine which feature

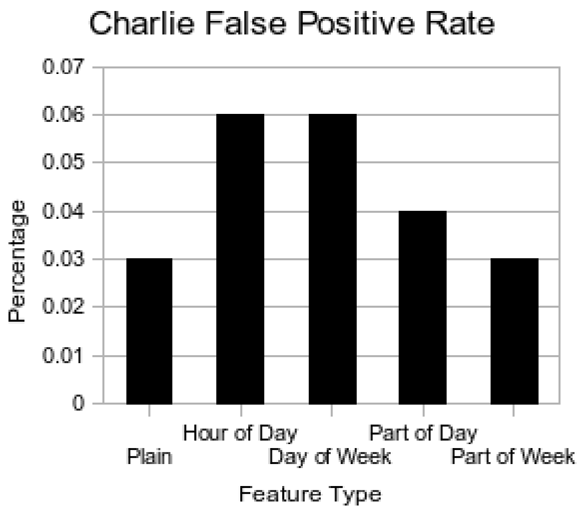

Figure 10: Charlie's rate of false positives across feature types.

type is most advantageous..

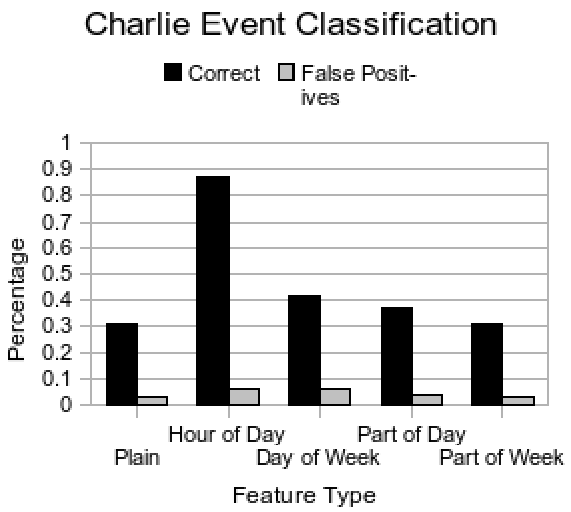

Figure 11: Overall classification rates for all features for Charlie.

For this data set, the other features, 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 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 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, the initial hypothesis of being able to garner additional information for training was borne out.

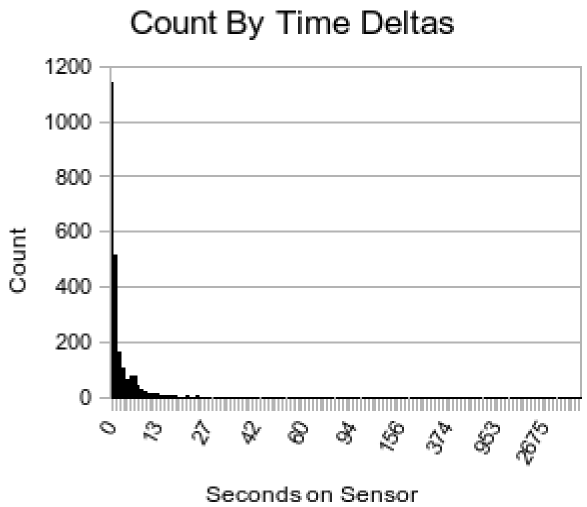

Figure 12: Count of lengths an individual spends on any sensor.

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

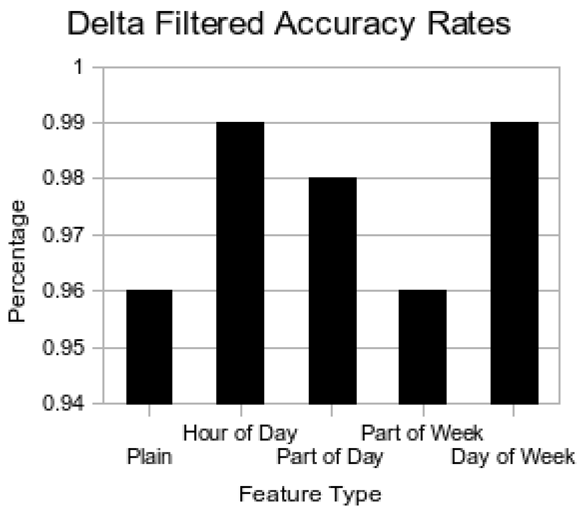

Figure 13: Delta filtered classification accuracy results.

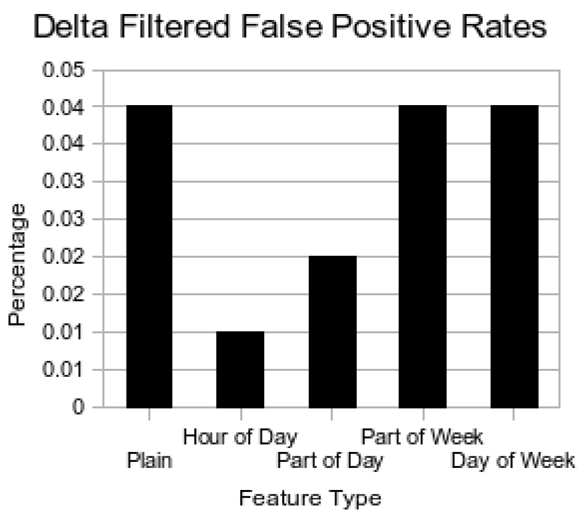

Figure 14: Delta filtered classification false positive results.

of the classifier. Note that with this filtered-data approach, sensor events with short durations will not be assigned a mapping to a specific resident. However, by combining this tool with one that tracks inhabitants through the space[7], only a handful of sensor events need to be classified as long as they have a high accuracy.

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 & 14. Once again, the hour-of-day performed the best, as it seems to give the naïve Bayesian classifier information that could be used to differentiate between resident behaviors within this data set.

6 Conclusions

In the interest of being able to identify a unique individual in a smart home, an approach leveraging a naïve Bayesian classifier was proposed. It was 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 inhabitants, 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 tool also lends itself to being coupled with tracking systems to identify people across a series of events by taking into account the likelyhoods at every stage of a person's behavior. Continued work combining simple tools like these are leading to very strong identification strategies.

7 Future Work

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.

Applying this classifier to a larger preference and decision engine is a must. 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 by NSF grant IIS-0121297.

References

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

[2] 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*, 2007.

[3] J. Krumm et al. Multi-camera multi-person tracking for easy living. In *Proc. 3rd IEEE Int'l Workshop Visual Surveillance*, pages 3–10, Piscataway, N.J., 2000. IEEE Press.

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

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

[6] S. S. Intille. Designing a home of the future. *IEEE Pervasive Computing*, vol. 1:80–86, Apr./June 2002.

[7] V. Jakkula, A. Crandall, and D. Cook. Knowledge discovery in entity based smart environment resident data using temporal relation based data mining. *IEEE Workshop on Temporal-Spatial Data Mining*, 2007.

[8] Vikramaditya Jakkula and Diane Cook. Using temporal relations in smart environment data for activity prediction. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.

[9] C. Lu, Y. Ho, and L. Fu. Creating robust activity maps using wireless sensor network in a smart home. In *Proceedings of the 3rd Annual Conference on Automation Science and Engineering*, Scottsdale, AZ, September 2007. IEEE.

[10] S. Moncrieff. Multi-model emotive computing in a smart house environment. *Journal of Pervasive and Mobile Computing*, vol. 3, 2007.

[11] R.J. Orr and G.D. Abowd. The smart floor: A mechanism for natural user identification and tracking. In *Proc. 2000 Conf. Human Factors in Computing Systems (CHI 2000)*, New York, 2000. ACM Press.

[12] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location support system. *Proc. 6th Ann. Int'l Conference on Mobile Computing and Networking*, pages 32–43, August 2000.

[13] 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 *Proc. of IEEE Int'l Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 393–403, July 2005.

[14] L. Snidaro, C. Micheloni, and C. Chivedale. Video security for ambient intelligence. *IEEE Transaction on Systems, Man and Cybernetics, part A*, Vol. 35:133–144, 2005.

[15] E. Munguia Tapia, S. S. Intille, and K. Larson. *Activity recognition in the home setting using simple and ubiquitous sensors*, volume LNCS 3001 of *Proc. PERVASIVE*, pages 158–175. Springer-Verlag, Berlin / Heidelberg, 2004.

[16] C.R. Wren and E.M. Tapia. Toward scalable activity recognition for sensor networks. *Hazas, M., Krumm, J., Strang, T. (eds.) LoCA 2006*, 3987:168–185, 2006.

[17] J. Yin, Q. Yang, and D. Shen. Activity recognition via user-trace segmentation. Jan 2007.