Modeling Human Behavior from Simple Sensors in the Home

Ryan Aipperspach, Elliot Cohen, and John Canny

Berkeley Institute of Design
University of California, Berkeley
Berkeley, CA 94720-1776 USA
{ryanaip, jfc}@cs.berkeley.edu, emcohen3@berkeley.edu

Abstract. Pervasive sensors in the home have a variety of applications including energy minimization, activity monitoring for elders, and tutors for household tasks such as cooking. Many of the common sensors today are binary, e.g. IR motion sensors, door close sensors, and floor pressure pads. Predicting user behavior is one of the key enablers for applications. While we consider smart home data here, the general problem is one of predicting discrete human actions. Drawing on Activity Theory, the Language-as-Action principle, and Speech understanding research, we argue that smoothed n-grams are very appropriate for this task. We built such a model and applied it to data gathered from 3 smart home installations. The data showed a classic Zipf or power-law distribution, similar to speech and language. We found that the predictive accuracy of the n-gram model ranges from 51% to 39%, which is significantly above the baseline for the deployments of 16, 76 and 70 sensors. While we cannot directly compare this result with other work (lack of shared data), by examination of high entropy zones in the datasets (e.g. the kitchen triangle) we argue that accuracies around 50% are best possible for this task.

1 Introduction

A number of research groups [8],[17],[9] and corporations [2],[4] have begun to study the role of computing in the digital home. There is not yet a consensus on the form that technology in digital homes should take or the purpose it should serve. Several groups, however, are considering the impact that small, inexpensive sensors might have on the home environment [1],[9]. Sensors can be used to support tasks such as activity recognition [9], health monitoring [13], and energy management [8]. In each of these cases, sensor data is used to determine the state of the home, making it possible to construct a more adaptive environment that responds to the needs of its inhabitants. There are several kinds of sensor analysis tasks including sensor fusion, interpretation, and prediction. We are most interested in the prediction task because it involves modeling of human behavior. Prediction from smart home data has been explored by several groups, especially for lighting and heating control [8]. Additionally, predictive behavior modeling can be used to build tutors for cooking or other everyday tasks. Behavior models can typically be used to recognize *anomalous* behavior as well such as deviations from routine, or skipped steps in Activities of Daily Living (ADL) for elders with onset dementia [21].

In this paper we present a very efficient behavior model for predicting future sensor outputs (and the user's location) from previous data. The method is scalable, works with a variety of sensor types, and is independent of the physical layout of the sensors. It can be trained in an *unsupervised* manner without requiring a configuration process or a room model. Setup consists solely of installing the sensors, putting them in training mode for a few weeks, and then starting prediction. The system continues to adapt its model from that time on. For us, it is also important to study models that are plausible from the perspective of our current understanding of human behavior.

We begin the paper by motivating the model we chose, introducing the "Language-As-Action" principle. The model itself is based on smoothed n-grams commonly used in language modeling. Then we explain the method in detail. We next describe the smart home datasets we used, which came from MIT and Georgia Tech. Then we present the results from running the model on the smart home datasets. This section includes analysis of the n-gram statistics, showing the classic power law or Zipf distribution commonly seen in speech and language data. We also discuss the limits to this kind of predictive model given the presence of high-entropy regions, like the "kitchen triangle". Finally, we discuss the implications of our results and future work.

2 Background

There has been a remarkable parallel evolution of a principle of "language as action." It was articulated first by the psychologist and educational theorist Lev Vygotsky [18] who along with Piaget remain as the two dominant figures in human learning research. Vygotsky also articulated various "genetic principles" governing human behavior. The principles imply that human behavior evolves at both a social and an individual scale. We found interesting support for Vygotsky's genetic principle in our smart home data, as we will discuss in the results section. The principle of language as action, is deeply embedded in the work of certain literary theorists, most notably Kenneth Burke [3] one of the most influential theorists of the mid-20th century. And most recently it has been fundamental to the work of the psychologist James Wertsch [19]. The crux of these theories are that language and human action are really the same thing. They are both "mediational means" or tools by which we achieve our ends. They exhibit structure and satisfy "grammars" (Burke's terminology). While the structure exists at many levels, there are strong similarities even at the most simple level – here we model smart home sensor data with language models that are normally used for words in a large corpus of text. We further show that the sensor outputs show the same fundamental statistics as texts (Zipf statistics). This is a far from obvious outcome – Zipf distributions are very "unnatural" in a statistical sense, say if one assumes that behavior is a result of a rational deliberation process. They are however, a universal trait of evolution (where they were first studied). In particular, they can arise from the evolution of behaviors – even simple behaviors such as walking around the house. Because of this deep connection, and because language and speech technology is one of the most heavily studied areas of human-machine interaction, we draw our behavior model directly from language modeling. For the latter, the state-of-the-art is a smoothed Markov or n-gram model (not to be confused with Hidden-Markov Models which are used for other tasks in speech processing) [5]. N-gram models are used in virtually every speech understanding system, and increasingly in information retrieval as well.

For the purposes of this paper, when we we refer to sensors we are explicitly considering simple "on/off" sensors such as motion detectors, status-reporting light switches, and appliance usage sensors. We consider these types of sensors for several reasons. First, we agree with the idea of "tape on and forget" [17] sensors that can be easily installed by end users with a minimal amount of configuration. We believe that systems using such sensors will be adopted more quickly than systems using complex sensors that require specialized installation. Simple sensors may also be seen as less invasive than complex sensors like cameras or microphones [17]. Additionally, simple sensors should be easier to integrate into the environment, an important consideration in light of the fact that some subjects describe their homes as seeming "dirty" when visible sensors are installed [1].

As an example of the specific need for local sensor event prediction, we consider the University of Colorado, Boulder, *Adaptive House*, which used sensors to automate lighting control. The *Adaptive House* faced the problem that lights only turned on after an inhabitant's motion in a room was detected, causing a perceptible lag in system responsiveness [8]. This problem was solved through the creation of a customized neural network designed to predict the state of the system two seconds into the future. While effective, the *Adaptive House* was customized to a specific home environment and prediction task, lacking the generality and scalability needed to satisfy the design goals of simplicity and ease in installation.

The MavHome project also focuses on creating intelligent homes through the use of prediction [10],[14]. However, information about data perplexity and running time is not available, making it difficult to compare our results directly.

3 Methods

We used the following five design goals in designing our prediction system[1]:

- **Probabilistic prediction.** The use of probabilistic methods makes it possible to associate a degree of confidence with each prediction and to consider

[1] These design goals are based on those presented by Tapia et al. for activity recognition [17].

a range of likely events, enabling the system to deal with the noise and ambiguity inherent in sensor data.

- **Model-based vs instance-based learning.** The incremental construction of models from training data makes it possible to build a predictor without the need to save all examples as raw data.
- **Sensor location and type independence.** Systems should operate effectively "even when the algorithm is never explicitly told the location and type of a particular sensor" [17], minimizing installation times and lowering barriers to adoption.
- **Real-time performance.** Any practical prediction algorithm must be able to make predictions in real-time.
- **Online learning**. Any system designed for long-term use must be able to adapt its model to support changes in inhabitant behavior over time.

In addition to these design goals, we found motivation in the similarity between streams of sensor data and language described previously. Language modeling algorithms often assume that languages are ergodic, having the property that the probability of any state can be estimated from a long enough history independent of earlier conditions [15]. One measure of this local structure is perplexity which, roughly speaking, is a measure of the number of words that might follow a particular word given its history. In the English language, perplexity can range from 20 for specialized subsets of the language to 247 for general American English [11]. The sensor data used in this project has perplexity ranging from 4 to 21, suggesting that predictive algorithms which work well in language modeling should work as well or better in short-term sensor event prediction. We chose to begin exploring this direction by using n-gram language models, mapping directly between sensor events in our system and words in language models. As we will discuss later, the actual distribution of sensor data supports the standard assumptions made by n-gram models.

The goal of language modeling is to calculate the probability of a word w_i given its history – that is, to compute $P(w_i|w_1, \ldots, w_{i-1})$. If a language is ergodic, then this probability can be estimated by $\widehat{P}(w_i|w_{i-n+1}, \ldots, w_{i-1})$, for a sufficiently large n. An n-gram language model can be used to calculate the maximum likelihood estimate of $\widehat{P}$ by counting word sequences in a set of training text:

$$\widehat{P}(w_i|w_{i-n+1}, \ldots, w_{i-1}) = \frac{C(w_{i-n+1}, \ldots, w_i)}{C(w_{i-n+1}, \ldots, w_{i-1})}, \quad (1)$$

where $C(\cdot)$ is the count of a given word sequence in the training text. The Hidden Markov Model Toolkit (HTK)'s language modeling tools [22] provide tools for collecting n-gram statistics. We based our system on these tools, augmenting them with code to make predictions based on such models. HTK provides several optimizations in collecting n-gram statistics, including a smoothing method incorporating *back off* and *Good-Turing* discounting as described in [6].

In any set of training data, it is unlikely that all possible sequences will be observed. However, it is unreasonable to assume that unobserved sequences are impossible. This dilemma can be overcome by "discounting" the probability of all observed sequences by some small amount and distributing the extra probability among the unobserved sequences. When using back off, the extra probability is distributed based on the likelihood of shorter sequences – An unobserved sequence $(w_{i-n+1}, w_{i-n+2}, \ldots, w_i)$ will receive higher weight if the shorter sequence $(w_{i-n+2}, \ldots, w_i)$ is common. In HTK, back off is implemented by calculating the probability $\widehat{P}(w_i|w_{i-n+1}, \ldots, w_{i-1})$ using the equation

$$\widehat{P}(w_i|w_{i-n+1}, \ldots, w_{i-1}) = \begin{cases} \alpha \cdot \widehat{P}(w_i|w_{i-n+2}, \ldots, w_{i-1}) & : \mathit{count} = 0 \\ d_C \cdot \frac{C(w_{i-n+1}, \ldots, w_i)}{C(w_{i-n+1}, \ldots, w_{i-1})} & : 1 \leq \mathit{count} \leq k \\ \frac{C(w_{i-n+1}, \ldots, w_i)}{C(w_{i-n+1}, \ldots, w_{i-1})} & : \mathit{count} > k. \end{cases} , \quad (2)$$

where α is the fraction of the discounted probability given to the unobserved sequence, d_C is the factor that discounts probability from observed sequences and *count* is the number of examples of the given sequence in the training data. For a full description of the implementation of smoothing in HTK, see [22].

The use of n-gram models fits the design requirements described above:

- **Probabilistic classification.** The counts used in n-gram models provide a probabilistic prediction of sensor events.
- **Model-based vs instance-based learning.** N-grams build a predictive model in the form of gram counts, and original data can be discarded as global statistics are accumulated.
- **Sensor location and type independence.** N-grams do not require any knowledge of the specific sensors being used or their location in the home. Since they consider common sequences in the data, they learn the structure of the data without requiring difficult or tedious system setup (although "good" sensor placement will still improve system performance).
- **Real-time performance.** The authors of [17] suggest that temporal models such as dynamic belief networks (DBNs) may not scale well to environments with hundreds of sensors. N-grams take advantage of temporal information through the use of simple and fast counting methods, allowing them to easily deal with large sets of data. Their empirical performance on sensor data will be discussed later.
- **Online learning.** The gram counts collected by n-gram models can be continually updated, allowing the system to adapt to changing patterns in user's routines.

4 Data Set

We are aware of few projects that have considered sensor data collected from non-laboratory home environments. While it is possible to generate data sets through simulation [14], it is important to validate algorithms on real data collected in complex, noisy environments [17]. We developed and tested our system primarily using data from the Georgia Institute of Technology *Aware Home* project [12], and we conducted additional tests using data from the Massachusetts Institute of Technology *House_n* project [17].

The Georgia Tech data set is a one year database of sensor events collected as part of the digital family portrait project [13]. In the project, the home of a single elderly resident was equipped with 16 in-floor pressure sensors that were triggered whenever someone walked over them. Since the sensors detect pressure in the floor, they can be completely invisible, making them good candidates for deployment in actual homes [1]. The layout of the residence and the sensors is shown in figure 1. Most of the time, the single resident was the only person in the home.

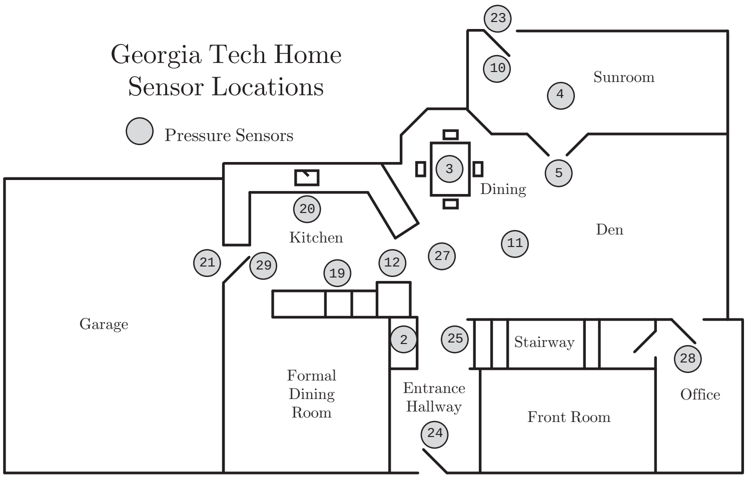

Fig. 1. Georgia Tech Floor Plan and Sensor Layout

The data was provided to us as a list of *(sensor, timestamp)* tuples. Because some sensors would repeatedly fire as the resident walked across them, we collapsed repeat firings into a single event. Additionally, we inserted `PAUSE` events into the data whenever the time between two sensor events was greater than some time t_{pause}, which was an input parameter. These pause events were inserted in order to model the difference between dwell spots (a sensor followed by a pause event) and paths through the home (a sequence of sensors without a pause event).

We also tested our system on data from the MIT *House_n* project [17]. The data set consists of two week segments of sensor data collected from two different single resident apartments. In the study, the apartments were instrumented with 76 and 70 “state-change” sensors, respectively. The sensors reported the status of numerous aspects of the homes, including doors being open or closed, appliances being in use or idle, and lights being on or off. The smaller amount of training data and the larger number and variety of sensors makes the *House_n* data set an interesting means of exploring the versatility of our algorithm.

Each data set was transformed into an ordered sequence of sensor labels and `PAUSE` events, and our goal was thus to predict the next sensor to be triggered given the sequence of sensors that were triggered recently.

5 Results

5.1 Data Analysis

The distribution of n-gram sequences in the Georgia Tech data set, shown in figure 2, is similar to a the Zipf distribution often seen in language. Zipf, or power law distributions are described by the relation $N_r \sim 1/r^a$, where r is the rank index of a particular sequence and N_r is the number of occurrences of that sequence. On a log-log plot, such distributions appear as a straight line, which can be seen in the right side of figure 2. Zipf distributions are indicative of "genetic processes", such as those described by Vygotsky [18]. In particular, evolutionary development of "populations" of species in biological genera [20], of city sizes [16], and, in this case, of behaviors has been shown to manifest a Zipf distribution.

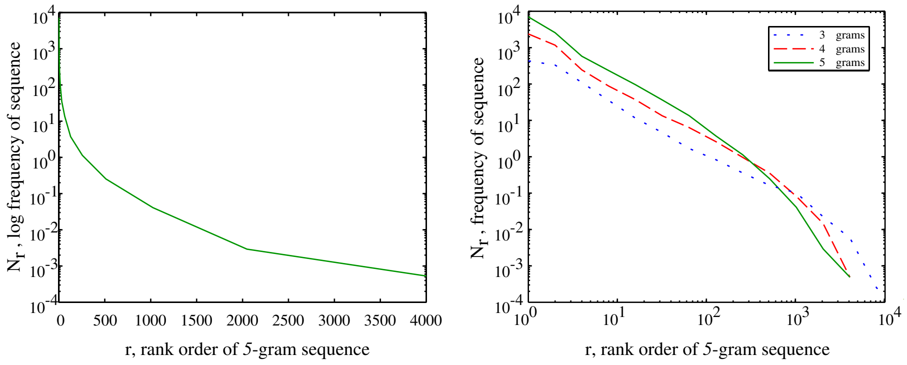

Fig. 2. Distribution of n-gram occurrence counts in the Georgia Tech data set on both linear (left) and logarithmic axes (right). r is the rank ordering of sequences, from most to least common, and N_r is the relative frequency of each sequence. The linear curve on the logarithmic axes is characteristic of a power law (Zipf) distribution.

Looking more closely at the data, table 1 shows the five most common *6*-grams in the Georgia Tech data set (with a sensor timeout, t_{pause}, of 5 minutes). The sequences seem to follow common pathways through the home. Additionally, they reflect a local structure that supports the ergodicity assumption behind n-gram language models. E.g., the local "inner sequence" $27, 3, 11, 5$ is visible in each of the last three sequences and is itself one of the most common *4*-grams.

This mapping between language and human behavior, evidenced through the Zipf distribution of movement sequences and the "local structure" of those sequences, supports the application of language modeling techniques to the modeling of human behavior.

Table 1. Most common *6*-grams in the Georgia Tech data set

Sequence						Percent of Total Observations
20,	19,	20,	19,	20,	19	2.26%
19,	20,	19,	20,	19,	20	2.17%
20,	12,	27,	3,	11,	5	1.72%
12,	20,	12,	27,	3,	11	1.56%
11,	5,	4,	PAUSE,	4,	5	1.26%

5.2 Model Performance

We first tested our system on the Georgia Tech data set, with and without the back off and smoothing optimizations as implemented in the HTK language modeling tools [22]. The optimizations had a significant impact on performance, as shown in table 2. The results are also significantly above baseline – completely random guessing would result in an expected accuracy of 6.25%[2].

Table 2. Effect of Back Off and Smoothing on the Georgia Tech Data Set. (The "top 2" and "top 3" results consider cases where any one of the top two or three most likely predictions is correct.)

	n-grams with Back Off & Smoothing	Simple *n*-grams
Percent correct	51%	38%
Percent correct (top 2)	67%	50%
Percent correct (top 3)	72%	62%

Table 3 shows the results of using the *n*-gram model to compute single-step predictions with back off and smoothing on the three data sets. The "G. Tech

[2] Note that it would be possible to make a somewhat smarter guess by picking a neighboring sensor rather than a random sensor, but this would violate the design requirement of sensor location independence since it would require knowledge about the layout of sensors within the home (and thus a more complicated system configuration process).

Limited" column includes the results of training the system on 2,000 events from the Georgia Tech data set, which is equivalent to the amount of data available in the MIT *House_n* data sets. The similarity between the results in "G. Tech Limited" and the two MIT data sets suggests that the lower performance on the MIT data relative to the full Georgia Tech data set is a function of the amount of training data available rather than of the larger number of sensors in the MIT installations. The n-gram model may be capable of accommodating the increased number and variety of sensors in the MIT data sets, but we cannot confirm this fact without the availability of more training data.

Table 3. N-gram model results

	G. Tech	G. Tech Limited	MIT 1	MIT 2
n-gram size	5	3	3	3
Percent correct	51%	44%	39%	43%
Percent correct (top 2)	68%	63%	47%	48%
Percent correct (top 3)	72%	68%	49%	49%
Perplexity	3.65	5.69	16.8	17.2
Number of sensors	16	16	76	70

5.3 Discussion of Results

When interpreting the predictive results of the n-gram model, it is important to consider the nature of the paths that residents take through the home. In many cases, the high entropy in the data may impose a limit on the ability to make predictions of future movement at the sensor level. If in a given situation there are a number of sensors that are equally likely to be the next sensor, then we cannot do better than make a random choice between them. As an example, we consider the Georgia Tech data set.

Many of the most common paths in the Georgia Tech data set occur within the "kitchen triangle" (sensors 12, 19, and 20 – see figure 3), which suggest the process of preparing a meal and moving between the sink, the stove, and the refrigerator. In fact, the top four n-grams shown in table 1 all include various movements among the kitchen sensors as a subsequence. This means that, given that the resident has triggered one of the three kitchen sensors, it is likely that her next movement will be to one of the other two. Choosing between these two sensors would suggest a maximum accuracy of 50%, which is approximately what the n-gram-based model achieves.

This level of accuracy is also useful for many possible applications. In energy management and lighting control systems, any increase in predictive performance will have an impact on system efficiency. Typical systems (e.g., [8]) make use of a

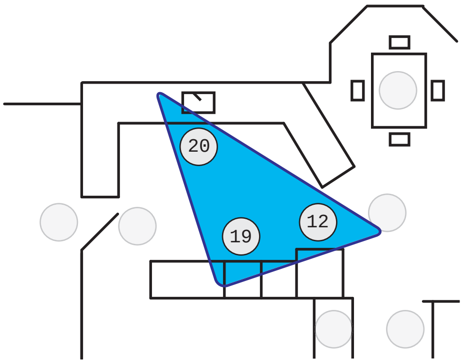

Fig. 3. "Kitchen triangle"

cost function that penalizes the system both for wasting energy (e.g., turning on lights or heating when no one is around) and for inconveniencing residents (e.g., not turning on lights when someone enters a room). Depending on the balance between these two penalties, a system can choose how cautious to be in reducing energy costs. For example, if a resident with a lighting control system begins to move around her home, a system with low predictive performance may turn on most of the lights in the house until the resident stops moving while a system with perfect performance could turn on only the lights along the resident's predicted path. Systems with performance between these two extremes can turn on lights along several of the most highly predicted paths (for example, using the top 2 or top 3 sensors as predicted in table 3, with incremental increases in predictive performance enabling them to move toward more optimal operation.

5.4 Performance

One key advantage of n-gram models is their speed. Because they are implemented primarily through counting, n-grams require very little processing time, especially when compared to complex methods such as DBNs. In our implementation, it took 48 seconds to construct a model of the one year Georgia Tech data set, which consisted of 134,000 data points. (All results are for a 1.6 GHz Pentium M system with 512 MB of RAM running Windows XP.) Prediction times for the three data sets are shown in table 4. Based on the observed per-sensor prediction time of 0.01 ms for *10*-gram models, the system should be able to support a deployment of 1,000 sensors while making predictions at a frequency of 100 Hz.

6 Future Work and Conclusion

As suggested by the close ties between human activity and language, predictive models used in language modeling can be applied successfully to behavior

Table 4. Average Time Per Prediction

	Georgia Tech	MIT Apartment 1	MIT Apartment 2
2-grams	0.0216 ms	0.129 ms	0.122 ms
5-grams	0.0610 ms	0.385 ms	0.356 ms
10-grams	0.174 ms	0.857 ms	0.813 ms
15-grams	0.345 ms	1.37 ms	1.34 ms
Number of Sensors	16	76	70
Prediction Time per Sensor (*10*-grams)	0.01 ms	0.01 ms	0.01 ms

modeling. N-grams provide a fast and accurate method for making single-step predictions on home sensor data, and we have argued that they achieve close to the best possible accuracy achievable on the task. However, n-gram models do not take into account higher level information such as task, activity or goal. Systems which integrate such high level information with low level analysis have been shown to be effective [7] in modeling human activity. We plan to explore building a similar hierarchical system on top of our existing framework.

The level of performance necessary in predictive systems will ultimately depend on the types of applications in which they are deployed. Additionally, measures of performance will include more than just accuracy and should be based on the actual impact predictions have on system behavior and their utility to household residents. Such systems must also take into account issues such cost of deployment, privacy, and error recovery methods. We plan to move quickly toward the implementation of applications in actual homes in order to assess the types of prediction errors that are the most problematic and to determine what level of performance is expected by users.

7 Acknowledgments

We would like to thank Leon Barrett for his help with an earlier iteration of this project. We would also like to thank James Rowan from Georgia Tech and Emmanuel Munguia Tapia from MIT for providing us with their data sets, without which this project would have been impossible.

References

1. J. Beaudin, S. Intille, and E. Tapia, “Lessons Learned Using Ubiquitous Sensors for Data Collection in Real Homes,” in *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI 2004).
2. B. Brumitt et al., “EasyLiving: Technologies for Intelligent Environments”, in *Handheld and Ubiquitous Computing*, September 2000.

3. Kenneth Burke, *Language as Symbolic Action,* University of California Press, 1966.
4. Intel Corporation, "Digital Home, Technology and Research at Intel," `http://www.cc.gatech.edu/fce/ecl/projects/dfp/index.html`
5. Frederick Jelinek, *Statistical Methods for Speech Recognition*, Cambridge, Massachusetts: MIT Press, 1997, p. 58.
6. S.M. Katz, "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recogniser," in *IEEE Transactions on Acoustic, Speech and Signal Processing*, 1987, vol. 35 no. 3. pp. 400-401.
7. L. Liao, D. Fox, and H. Kautz, "Learning and Inferring Transportation Routines," in *Proceedings of AAAI-04*, 2004.
8. M. Mozer, "Lessons from and Adaptive House", in *Smart Environments: Technologies, protocols, and applications*, D. Cook and R. Das Eds. Hoboken, NJ: J. Wiley and Sons, 2005, pp. 273-294.
9. M. Philipose, et al., "Inferring Activities from Interactions with Objects," in *Proceedings of the Conference on Pervasive Computing*, October 2004, pp. 50-57.
10. S. Rao and D. J. Cook, "Identifying Tasks and Predicting Actions in Smart Homes using Unlabeled Data", in *Proceedings of the Machine Learning Workshop on The Continuum from Labeled to Unlabeled Data*, 2003.
11. S. Roukos, "Language Representation," in *Survey of the State of the Art in Human Language Technology*, R.A. Cole et al., Eds. Center for Spoken Language Understanding CSLU, Carnegie Mellon University, 1995.
12. J. Rowan, Digital Family Portrait project, `http://www.cc.gatech.edu/fce/ecl/projects/dfp/index.html`.
13. J. Rowan and E.D. Mynatt, "Digital family portraits: Providing peace of mind for extended family members," in *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI 2001), Seattle, Washington: ACM Press, 2001, pp. 333-340.
14. A. Roy, et al., "Location Aware Resource Management in Smart Homes", in *Proceedings of the Conference on Pervasive Computing*, 2003.
15. C.E. Shannon, "A Mathematical Theory of Communication," in *The Bell System Technical Journal*, vol. 27, 1948. pp. 379-423.
16. H.A. Simon, "On a class of skew distribution functions," in *Biometrika*, vol. 42, 1955. pp. 425-440.
17. E. Tapia, S. Intille, and K. Larson, "Activity recognition in the home setting using simple and ubiquitous sensors," in *Proceedings of PERVASIVE 2004*, vol. LNCS 3001, A. Ferscha and F. Mattern, Eds. Berlin Heidelberg: Springer-Verlag, 2004, pp. 158-175.
18. L. S. Vygotsky, *Mind in Society: The development of higher psychological processes.* Edited by Michael Cole, Vera John-Steiner, Sylvia Scribner and Ellen Souberman, Harvard University Press, 1978.
19. J. Wertsch, *Mind As Action,* Oxford University Press, 1998.
20. J. C. Willis and G. U. Yule, "Some statistics of evolution and geographical distribution in plants and animals, and their significance," in *Nature*, vol. 109, 1922. pp. 177-179.
21. D.H. Wilson and C. Atkeson, "Simultaneous Tracking and Activity Recognition (STAR) Using Many Anonymous, Binary Sensors," in *Proceedings of PERVASIVE 2005*, Munich, Germany, May 2005.
22. S. Young, et al., *The HTK Book*, Microsoft Corporation and Cambridge University, 3.2.1 edition, 2002.