

Structural Learning of Activities from Sparse Datasets

Fahd Albinali¹, Nigel Davies^{1,2}, and Adrian Friday¹

¹ University of Arizona, Tucson AZ 85704, USA,
albinali@cs.arizona.edu

² Lancaster University, Lancaster LA1 1WB UK,
{nigel,adrian}@comp.lancs.ac.uk

Abstract. A major challenge in pervasive computing is to learn activity patterns, such as bathing and cleaning from sensor data. Typical sensor deployments generate sparse datasets with thousands of sensor readings and a few instances of activities. The imbalance between the number of features (i.e. sensors) and the classification targets (i.e. activities) complicates the learning process. In this paper, we propose a novel framework for discovering relationships between sensor signals and observed human activities from sparse datasets. The framework builds on the use of Bayesian networks for modeling activities by representing statistical dependencies between sensors. We optimize learning Bayesian networks of activities in 3 ways. Firstly, we perform multicollinearity analysis to focus on orthogonal sensor data with minimal redundancy. Secondly, we propose Efron’s bootstrapping to generate large training sets that capture important features of an activity. Finally, we find the best Bayesian network that explains our data using a heuristic search that is insensitive to the exact ordering between variables. We evaluate our proposed approach using a publicly available data set gathered from MIT’s PlaceLab. The inferred networks correctly identify activities for 85% of the time.

1 Introduction

A central theme in pervasive computing is the need to build systems that use low-level sensor data to identify the Activities of Daily Living (ADLs) (such as cleaning or bathing). Recently, there has been significant progress in this area including (a) deploying real systems that extract information from a variety of sensors [1], (b) developing and extending machine learning approaches that model activities including Bayesian [2], Markov [3], Neural [4], and rule-induction [5] classifiers and (c) studying specific domains to identify relevant context information which in turn can be used to support activity recognition [6].

The vision of developing systems that recognize ADLs has been driven by numerous compelling applications. For example, in the medical field, professionals believe that changes in the activities of daily living such as eating, showering etc.

are important in the detection of emerging medical conditions and that monitoring the daily activities of people can reduce the number of low risk cases that come into hospitals – thereby reducing the financial burden on health systems and allowing more focus on patients at higher risk [6].

A major challenge in learning ADL patterns is the lack of large training sets that reflect accurate aspects of an activity. For instance, existing deployments such as the PlaceLab generate thousands of sensor readings along with only tens of instances of activities. Such sparse data cannot be used to learn a complete model of an activity. However, such sets contain important information that may reflect sensor patterns during activities. The goal is therefore to separate important sensor signals from noisy ones and to identify true relationships from spurious ones.

In this paper, we propose a novel framework for discovering relationships between sensor signals and observed human activities from sparse datasets. The framework builds on the use of Bayesian networks for modeling activities by representing statistical dependencies between sensors. We extend earlier efforts in three fundamental ways. Firstly, we perform multicollinearity analysis to focus on orthogonal features of our data (i.e. features with minimal redundancy). The goal here is to eliminate aspects of the data that do not contribute additional information. Secondly, we propose Efron’s bootstrapping to generate large training sets that capture important features of each activity. Bootstrapping is a non-parametric technique that resamples with replacement from the original data. The underlying intuition is that bootstrap runs will emphasize consistent features in our original data thereby facilitating learning. Finally, we find the best Bayesian network that explains our data using a heuristic search that is insensitive to the exact ordering between variables. We specifically eliminate the bias due to the ordering between consecutive variables. We evaluate our proposed approach using a publicly available data set gathered from MIT’s PlaceLab. The inferred networks correctly identify activities for 85% of our validation set.

2 Background

Most work on recognizing activities has centered on using statistical methods to build relationships between observed sensor data and ADLs. For example, the MARC home project [7] used mixture models and hierarchical clustering to detect breakfast and dinner preparation activities in the kitchen. Both activities were detected with high accuracy. Naïve Bayesian classifiers were used by researchers at MIT [1] to detect ADLs in home settings using *environmental state-change sensors*. For 13 ADLs, the percentage of time that an activity was correctly detected peaked at 31%. The Guide project [8] used unsupervised methods (that do not require activity labeling) to detect activities using RFID tags placed on objects. This method relied on "large quantities of reliable sensed data about activities" and achieved good recognition accuracy. More recently, Proactive Activity Toolkit (PROACT) [2] used *dynamic Bayesian networks* and RFID tags to build ADL models. Subjects were given task sheets for 14 differ-

ent activities that were performed using a glove equipped with an RFID reader. When the reader failed to capture a particular touch, subjects were asked to retouch objects. PROACT successfully inferred that an activity occurred 88% of the time. Finally, Doctor et al. [5] used fuzzy rule induction and genetic programming to construct rule sets for activities from sensor data. This approach examined 10 different activities and produced 306 rules to predict ADLs.

Many factors contribute to the difficulty in recognizing ADLs including: first, ADLs may vary in their frequency of occurrence from once a week to a number of times per day (e.g. washing clothes vs. going to the bathroom). The infrequency of many ADLs prevents the collection of sufficiently large data sets with balanced ADL entries. This in turn complicates the task of learning ADL patterns.

Second, human action is not always purposeful [9] and therefore observable action (that is captured by sensors) may not correlate to particular activities. This increases noise in ADL data sets and makes them difficult to analyze.

Third, many ADLs consist of logical sequences of sub-activities (e.g. cooking includes chopping, frying, serving etc.) and typically involve the firing of complex combinations of sensors. However, many ADL recognition methods do not explore different structural dependencies between sensors that in turn reduces model accuracy.

In the remainder of this paper, we present three techniques to address the above challenges. Section 3 discusses our approach to generating small feature vectors with minimal redundancy from raw sensor data. Section 4 describes bootstrapping as a method to generate large activity samples that emphasize important aspects of each activity. We then discuss how to learn Bayesian networks and present a heuristic search approach in section 5. Section 6 outlines our algorithm. Section 7 evaluates our approach. Finally, we conclude and discuss future work in section 8.

3 Feature Encoding and Selection

Proximity to objects has been the most critical aspect in modeling activities. Activations of proximity sensors are typically encoded to incorporate temporal information such as quantitative relationships (e.g. stove sensor fired during cooking) or qualitative relationships (e.g. the shower sensor fired after the bathroom light sensor). Intille et al. shows that low-order temporal relationships between sensor activations are sufficient for recognizing ADLs using naïve Bayesian classifiers [10].

We use a similar approach to encode our data. First, we discretized sensor readings based on fixed thresholds. A possible criticism of discretization is that it penalizes sensors whose range of variation is small: because with a fixed threshold, we would not capture small changes. This turned out to be not critical because most of our sensor readings can be interpreted in a discrete manner. For instance, although current sensors measure continuous current flows, we easily distinguished 2 levels of current flow that indicate that a device is either on or off. Second, we encode consecutive sensor readings into a single feature vector where

$F(S_i, S_j) = 1$ implies that sensor S_i fired before sensor S_j and $F(S_i, S_j) = 0$ implies that sensor S_i did not fire before sensor S_j .

In a setup of N sensors, we have $N \times N$ features that can be included in our model and therefore reducing the number of features is critical. We use 2 approaches to reduce the number of features for an activity. First, we only consider the features that fire in our training data during the activity. Second, we perform a multicollinearity analysis on the remaining features to discard those that are linearly correlated.

Multicollinearity analysis describes a statistical method to determine if columns of a matrix behave in a correlated manner. Specifically, we apply eigen decomposition on the correlation matrix of our features to determine linearly dependent features. This splits the data into orthogonal eigen vectors where associated eigen values reflect the magnitude of each vector. Ideally, all eigen values should be large reflecting that all orthogonal vectors are important and that there is minimal redundancy. The presence of small eigen values points to multicollinearities in the data (i.e redundancy).

Based on the eigen values, we choose the set of features with minimal redundancy (i.e. the most linearly uncorrelated). The intuition is that features that change together do not contribute any additional information to the learning process, instead they complicate the search for a solution and therefore such features should be eliminated. By applying this concept, we reduced our feature set without losing critical information.

4 Bootstrapping for Model Induction

Unlike typical applications of learning from data, sensor data from pervasive environments pose unique challenges. Firstly, pervasive environments are equipped with hundreds of sensors that are sampled at high rates. This generates large feature sets with many readings that can easily suppress the true signal of an activity. Secondly, there is a clear imbalance between the frequencies of activities and the amount of sensor readings that pervasive setups generate. Such sparse data cannot be used to learn a complete model of an activity and can easily skew model parameters to extreme values.

To overcome the above challenges, we propose the use of Efron’s non-parametric bootstrapping for data generation and subsequently model induction. The idea behind bootstrapping is that if an activity sample is a good approximation of the true activity, a sampling distribution can be estimated by generating a large number of new samples (called pseudo samples) by randomly sampling with replacement from the original data set. Therefore, if sensor S fires consistently within a particular activity A , then we should be able to detect that in our bootstrap runs. If only a small fraction of our bootstrap samples reflect the presence of S in A then probably S is unrelated to A .

We bootstrap both training and validation samples of our features for all activities. Using the generated data, we invoke the search algorithm to learn a

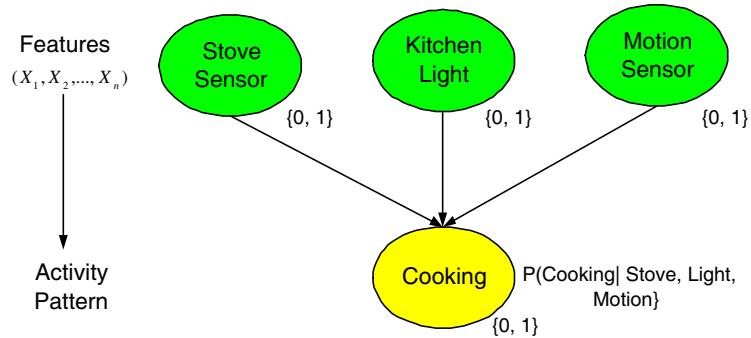


Fig. 1. Mathematical Structure of a Bayesian Network

Bayesian network for each activity. We then validate our results with respect to the bootstrap runs of our validation samples.

5 Learning Bayesian Networks

We summarize learning of Bayesian networks from data. For a more detailed exposition, we refer the reader to [11].

Consider a finite set of independent discrete random variables $X = X_1, \dots, X_n$ where each variable X_i may take on values from a finite set. A Bayesian network is a Directed Acyclic Graph (DAG) that describes relationships of probabilistic dependency between the above variables. More formally, a Bayesian network BN is a pair $\langle G, \theta \rangle$. G is a DAG whose nodes represent the random variables in X and whose edges represent probabilistic dependencies between the variables. θ represents the parameters of the Bayesian network including estimates of the prior distributions for root nodes and estimates of the conditional distributions for each edge. Therefore, a learned Bayesian network defines a unique joint probability distribution over X given by:

$$P_{BN}(X_1, \dots, X_n) = \prod_{i=1}^n P_{BN}(X_i | \text{parents}(X_i))$$

Figure 1 shows the mathematical structure of a node in a Bayesian Network.

Given some training data $D = d_1, \dots, d_m$ where d_i represent an assignment to the discrete random variables in our model, the problem of learning a Bayesian network involves two components: a scoring metric and a search procedure. The scoring metric measures the goodness-of-fit of the learned network to the training data D . This metric has to possess 3 properties. First, structure equivalence that means that for any two networks with identical structures, the score must be the same. Second, the score should be decomposable which means it can assess

the exact gain or loss due to adding, removing or changing the direction of an edge in a network. Finally, the score should have a closed form. The second component of the learning algorithm is a search procedure for the network with the highest score which is known to be NP-hard [12]. Instead, heuristic searches are typically used such as hill-climbing to find approximate solutions.

It is important to emphasize that the network structures that are discovered are sensitive to the search algorithm. For instance, hill-climbing techniques will often produce a set of networks that are very similar in structure as the algorithm converges to a solution. This is a serious limitation since we run the risk of over-restricting our search for dependencies among variables. Next, we describe a novel technique that performs a polynomially bounded search of networks that have fundamentally different structures.

6 Efficient Structural Learning of Activities

A basic method for learning a Bayesian network uses K2 hill-climbing [13] where arcs from a fixed ordering of variables (i.e. nodes) are added to each node to maximize the probability that the network is correct, given the data using Bayes formula:

$$P(N|D) = \frac{P(N)P(D|N)}{P(D)}$$

The probability of the data D , given our network, N , assumes a Dirichlet prior which is given by Cooper and Herskovits [13].

A major limitation of the above approach is sensitivity to the order of introducing arcs into the model. For instance, the impact of an important dependency between two variables can be suppressed by less critical dependencies because they were explored earlier in the list of variables. It should be noted that performing an exhaustive search for all possible networks is infeasible because searching for an optimal network is NP-hard.

Instead, we circumvent the limitation imposed by ordering by using a greedy search algorithm that is insensitive to ordering between consecutive nodes. Our algorithm examines the networks that are generated by the dihedral groups of our features. A dihedral group of n features is a set of permutations that include n rotations and n reflections of the features. Figure 2, shows the dihedral groups for 3 features. The intuition behind using these permutations is that rotations allow each sensor to be entered at least once as the first element in the learner thereby having the highest impact. Reflections switch the ordering of every 2 consecutive features thus eliminating the bias due to consecutive node ordering. More importantly, for n features we have $2n$ dihedral permutations which sets a polynomial upper bound on the number of networks to explore.

For each permutation, we use K2 hill-climbing [13] to learn the best network structure from independent bootstrap samples. We then choose the network with the highest score as the best classifier for an activity. Details of the algorithm are given in figure 3.

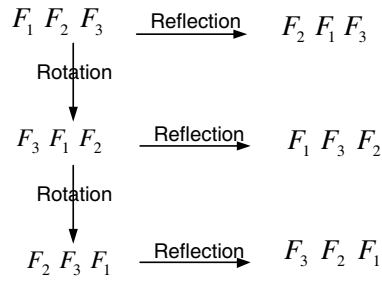


Fig. 2. Dihedral Permutations of 3 Features

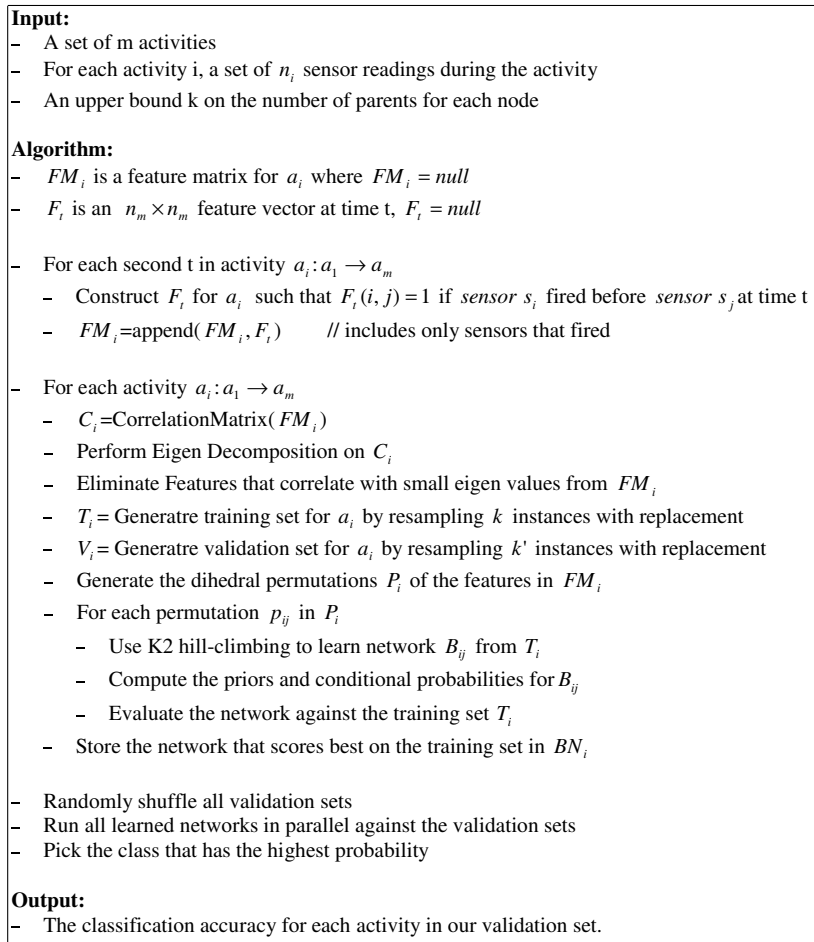


Fig. 3. Structural Learning Algorithm

7 Evaluation

We wanted to evaluate our approach using data sets from realistic settings where users perform everyday activities in environments that resemble their homes. Users should be able to multi-task their activities, determine their sequence and pace, interact with other people and engage in complex behaviors without interruptions. More importantly, we wanted to minimize the bias in our data sets that may result from imposing specific activities on users or installing sensors in ways that improve recognition of specific activities.

We found one publicly available dataset that fit our criteria from MIT’s PlaceLab [14]: a live-in laboratory in Cambridge, MA. The PlaceLab data was recorded on March 2005 for 4 consecutive hours where a researcher performed a set of common household activities. The researcher determined the type of activities, their sequence, their pace and whether they overlap. As the researcher conducted the activities, data was sampled from approximately 300 sensors such as current flow sensors and switch sensors. 9 infrared cameras and 9 color cameras were used to record activities throughout the space. Subsequently, trained annotators used the video feeds to annotate the sensor data with appropriate activity labels. This approach has at least two advantages: (1) users are not required to annotate their own activities and therefore have a more realistic experience in the space. (2) With video records, annotators can easily label activities at different granularity (e.g. a user is chopping vegetables and cooking). More information about the data set and its format can be found in [14].

7.1 Methodology

In our analysis, we looked at 24 different activities. 21 activities had at least 2 samples and therefore we were able to run independent bootstraps for training and validation data. 3 activities had 1 sample that we used for bootstrapping both our training and validation data.

We considered data from 286 sensors including: switch sensors that detect on and off events such as doors being opened or closed, light sensors that measure the degree of illumination in different areas, gas sensors that measure amount of gas flow to the water heater and stove burners, current sensors that detect current flows in 43 different circuits, water sensors that measure the amount of water flow for different faucets and toilets and MITes sensors that measure the movement of different objects.

The raw sensor data is initially preprocessed to obtain feature vectors of the form described in section 3. These feature vectors reflect sensor activity during every second of an activity. Multicollinearity analysis is then performed on the discrete features to eliminate linearly dependent features. The number of features is selected such that at least 90% of the variability in the data is maintained with an upper bound of 25 features per activity for computational reasons. This resulted in an average of 15 features per activity.

Using 2 sets of independent samples for each activity, we applied non-parametric bootstraps to generate 100 instances for training and 20 instances for validation

per activity. The training samples are given to our search algorithm that finds the best Bayesian network that explains our data. The search procedure uses K2 hill-climbing with restarts that examine the dihedral groups of our features. For 24 activities, the search algorithm generated 24 Bayesian networks.

Table 1. Percentage of time activity detected

<i>Activity</i>	Examples	Features	BNT	BNT-1FT	BNT-5FT	BNT-RS	RG
Cleaning a surface	7	15	88%	85%	28%	88%	4%
Cleaning	3	13	88%	69%	31%	82%	4%
Dishwashing	3	12	85%	69%	15%	38%	4%
Disposing Garbage	3	6	83%	52%	17%	67%	4%
Drinking	7	22	92%	84%	56%	20%	4%
Drying Dishes	3	19	88%	83%	38%	13%	4%
Eating a meal	3	23	94%	91%	43%	94%	4%
Handwashing Dishes	13	24	94%	90%	52%	75%	4%
Laundry	2	17	89%	78%	76%	97%	4%
Listening to music	1	22	92%	88%	58%	89%	4%
Making the bed	4	8	67%	57%	22%	78%	4%
Meal preparation	6	16	91%	85%	45%	91%	4%
Measuring	3	11	90%	80%	10%	90%	4%
Mixing and Stirring	9	17	92%	92%	50%	29%	4%
Mopping	4	25	93%	85%	81%	19%	4%
Preparing Food	2	9	75%	58%	8%	67%	4%
Preparing Meal	3	18	67%	71%	38%	95%	4%
Putting away laundry	1	14	87%	80%	27%	33%	4%
Retrieving ingredients	10	11	83%	74%	18%	92%	4%
Sweeping	3	7	71%	57%	29%	71%	4%
Using a computer	10	21	96%	88%	46%	95%	4%
Using a phone	6	3	68%	67%	33%	67%	4%
Washing Ingredients	1	18	90%	86%	50%	90%	4%
Watching Movies	3	5	80%	60%	24%	80%	4%
Mean	4.5	15	85%	76%	37%	70%	4%

7.2 Accuracy Analysis

To assess the accuracy of our approach, we randomly shuffled the activities in our validation set. Each activity had 20 instances for a total of 480 activities. We gave the feature sequences of each activity (without segmenting between activities) to our inference software that assessed the 24 Bayesian networks in parallel. A history of 10 seconds is used in setting the nodes of the Bayesian networks.

The inference software returns the most likely activity (for each second) based on a stringent criteria where a valid classification should have a probability that

is at least double the probabilities of other classifiers. Therefore a classification with probability 0.9 and 0.6 from BN1 and BN2 is inadmissible.

Table 1 shows the results for 24 ADLs in the column labeled BNT. Our networks correctly inferred that an activity occurred 85% of the time. This is significantly better than the random guess case (shown in table 1 in the RG column) which would yield an average of 4% in terms of classification accuracy. Our best performance was 96% for 'using a computer' which simply involved features of sensors located in the office area. This activity had a unique spatial locality that did not overlap with other activities. The worst performance was for 'preparing a meal' and 'making the bed'. The former activity significantly overlapped with 'preparing food' with respect to the underlying features. The latter activity generated a small number of sensor activations and was detected towards the end of the activity.

7.3 Robustness Analysis

We performed a number of tests to analyze the robustness of our procedure. We examined one particularly interesting hypothesis, namely whether accuracy increases as the number of features in our model increases. In classification problems, adding irrelevant features can degrade performance. We performed a regression fit between accuracy and the number of features followed by a hypothesis test. We found strong evidence that an increase in the number of features leads to improved accuracy ($p < 0.0001$). This result comes at no surprise, since we carefully selected orthogonal features to include in our model using multicollinearity analysis.

We also tested the robustness of our analysis to the addition of linearly dependent features. For each activity, we randomly selected 1 and 5 features with collinearity to the original feature set. We then added those features to our training and validation data and retrained our models. Table 1 shows the results of adding 1 and 5 features in the columns BNT-1FT and BNT-5FT respectively. As expected, the average classification accuracy dropped from 85% to 76% with 1 added feature and to 37% with 5 added features. 'Preparing Meal' was an exception where accuracy slightly increased by adding 1 feature. In this particular instance, we randomly picked the second least correlated feature to add. This feature still explained part of the variability in our data and therefore led to a slight improvement. However, the results clearly validate our methodology in selecting important features to include in our learning procedure.

Finally, we wanted to test our search procedure (that uses the dihedral permutations) against standard K2 hill-climbing with random restarts where the order by which arcs are added to a particular node from other nodes is random. The results of the randomized algorithm are shown in table 1 in the column labeled BNT-RS. Our approach outperformed the randomized algorithm by an average of 15% across all activities. However, this result is not consistent for all activities. In several instances, the randomized algorithm arrived at a better solution such as 'Preparing Meal'. This came at a cost where accuracy dropped for another overlapping activity namely 'preparing food'. More interestingly, the

randomized algorithm gave identical results to our algorithm for many activities which may suggest that order between consecutive features is not critical for those activities. However, we emphasize that both algorithms start with identical and orthogonal features which demonstrates the importance of our initial selection criteria and may explain why both algorithms arrived at similar conclusions.

8 Conclusion

This paper proposes a novel methodology for modeling activities from sparse datasets. Our work addresses a fundamental problem that arises in practical deployments: the imbalance between the number of activities and the number of sensor readings. Whereas previous efforts attempt to learn directly from sparse data, we focus on overcoming the sparsity of the available data. Initially, multicollinearity analysis eliminates redundant information in our data. Efron's bootstrapping is then used to generate training and validation sets from independent samples. Finally, we find the best Bayesian network that explains our data using a heuristic search that is insensitive to the exact ordering between consecutive variables. Our networks achieve an average accuracy of 85% and perform more consistently than hill-climbing with random restarts across 24 different activities.

It is important to note that our approach does not necessarily learn a complete model of an activity. However, without our proposed optimizations it would be difficult or perhaps impossible to learn patterns from sparse data. Moreover, many statistical assumptions will fail because of the lack of sufficiently large data sets. Instead, our approach emphasizes consistent aspects of the data and tries to search for a consistent feature profile of an activity.

Our plans for future work include exploring the causality relationships between the nodes in our Bayesian networks. A Bayesian network captures the probabilistic dependencies between a set of variables. This may or may not reflect causal relationships between the variables involved. We are interested in identifying pairs of nodes that have a causal relationship between them to attach meaning to our networks and to engage users in more intimate interactions. For example, users may be able to ask why the system incorrectly classified an activity and then attempt to correct system behavior by inducing some bias.

References

1. Tapia, E., Intille, S., Larson, K.: Activity recognition in the home setting using simple and ubiquitous sensors. LNCS Springer-Verlag **3001** (2004)
2. Philipose, M., Fishkin, K.P., Perkowski, M., Patterson, D.J., Fox, D., Kautz, H., Hahnel, D.: Inferring activities from interactions with objects. *IEEE Pervasive Computing* **3**(4) (2004) 50–57
3. Oliver, N., Horvitz, E., Garg, A.: Layered representations for recognizing office activity. *The Fourth IEEE International Conference on Multimodal Interaction* (2002) 3–8

4. Mozer, M.C.: The neural network house: An environment that adapts to its inhabitants. AAAI Spring Symposium (1998) 110–114
5. Doctor, F., Hagaras, H., Callaghan, V.: An intelligent fuzzy agent approach for realising ambient intelligence in intelligent inhabited environments. IEEE Trans. On Systems, Man and Cybernetics **35**(1) (2005) 55–65
6. Dalal, S., Alwan, M., Seifrafi, R., Kell, S., Brown, D.: A rule-based approach to the analysis of elders activity data: Detection of health and possible emergency conditions. AAAI Fall 2005 Symposium (2005)
7. Barger, T., Alwan, M., Kell, S., Turner, B., Wood, S., Naidu, A.: Objective remote assessment of activities of daily living: Analysis of meal preparation patterns. Poster Presentation (2002) Medical Automation Research Center, University of Virginia Health System.
8. Philipose, M., Fishkin, K., Fox, D., Kautz, H., Patterson, D., Perkowski, M.G.: Towards understanding daily life via auto-identification and statistical analysis. UbiHealth Workshop. UbiComp (2003)
9. Mises, L.: Human Action: A Treatise on Economics. Fox & Wilkes (1966)
10. Intille, S., Larson, K., Tapia, E., Beaudin, J., Kaushik, J.: Using a live-in laboratory for ubiquitous computing research. LNCS Springer-Verlag (2006)
11. Pearl, J.: Graphical models, causality and intervention (1993)
12. Chickering, D.: Learning bayesian networks is np-complete (1995)
13. Cooper, G.F., Herskovits, E.: A bayesian method for the induction of probabilistic networks from data. Machine Learning **09**(4) (1992) 309–347
14. Intille, S.S., Larson, K., Beaudin, J.S., Nawyn, J., Tapia, E.M., Kaushik, P.: A living laboratory for the design and evaluation of ubiquitous computing technologies. In: CHI '05: CHI '05 extended abstracts on Human factors in computing systems, New York, NY, USA, ACM Press (2005) 1941–1944