

Anomaly Detection Using Temporal Data Mining in a Smart Home Environment

V. Jakkula and D.J. Cook

Washington State University
EME 121 Spokane Street
Pullman, WA 99164, USA
Office: 509-335-4985
Fax: 509-335-3818
Email: cook@eecs.wsu.edu

Summary

Objectives: To many people, home is a sanctuary. With the maturing of smart home technologies, many people with cognitive and physical disabilities can lead independent lives in their own homes for extended periods of time. In this paper, we investigate the design of machine learning algorithms that support this goal. We hypothesize that machine learning algorithms can be designed to automatically learn models of resident behaviour in a smart home, and that the results can be used to perform automated health monitoring and to detect anomalies.

Methods: Specifically, our algorithms draw upon the temporal nature of sensor data collected in a smart home to build a model of expected activities and to detect unexpected, and possibly health-critical, events in the home.

Results: We validate our algorithms using synthetic data and real activity data collected from volunteers in an automated smart environment.

Conclusions: The results from our experiments support our hypothesis that a model can be learned from observed smart home data and used to report anomalies, as they occur, in a smart home.

Keywords: machine learning, smart homes, anomaly detection, temporal relations

1 Objectives

By 2040, a projected 26% of the US population will be 60+ and at least 45% of the populations of Japan, Spain and Italy will be 60 or older by then [1]. Approximately 13% of these older adults suffer from dementia and related disabilities. Given the costs of nursing home care and the importance individuals place on remaining in their current residence as long as possible, use of technology to enable individuals with cognitive or physical limitations to remain in their homes longer should be more cost effective and promote a better quality of life. As a long-term outcome of this investigation we expect to develop and to offer the community smart environment technologies with data mining and machine learning algorithms that can effectively perform a variety of health monitoring and intervention strategies.

We define a smart environment as one that collects data about the residents and the environment in order to adapt the environment to the residents and meet the goals of safety, security, cost effectiveness, and comfort. In an environment that is equipped with sensors to detect motion, temperature, and other conditions, sensed events can be captured and associated with a time stamp. The history of observed events reflects activities that occur in the environment and can be used to discover frequent recurring activity patterns [2], to recognize activities of daily living [3], to identify suspicious states [4], and to predict resident actions [5]. While researchers argue that space and time play essential roles in everyday lives [6], this is the first such study which incorporates time interval information into a health monitoring algorithm. These time intervals offer additional information about the relationships between timings of activities that improves the performance of health monitoring tasks such as anomaly detection.

Allen [7] suggested that it is more effective to describe activities using time intervals rather than time points, and defined thirteen relations that comprise a temporal logic. We refine Allen's temporal logic for use in analyzing smart environment data, and apply it to the task of anomaly detection. While other methods treat each event as a separate entity (for example, turning on a lamp and later turning off the same lamp), our interval-based analysis considers these two events as members of the same activity and therefore belonging to the same time interval. Each interval is expressed in terms of start time and end time values. As a result, temporal relationships between such intervals can be identified and used to perform critical anomaly detection. By recognizing that many activities in a smart environment have a distinct beginning and end with an associated time span, we can reason about temporal relationships between activities that regularly occur. For example, a data mining algorithm may note that a smart environment resident often makes popcorn during the same time interval that they are watching a movie. If popcorn has not been cooked by the time the movie is over, this may be considered an anomaly.

The objective of this study is to determine if anomalies can be effectively detected in smart home data using temporal data mining. Specifically, we introduce a temporal representation that can express frequently-occurring relationships between smart environment events. We then use the observed history of events to determine the probability that a particular event should or should not occur on a given day, and report as an anomaly the presence (or absence) of highly-likely events. To validate the approach, we test the algorithm on synthetic data as well as real data collected from a smart environment. We discuss the implications of this work for health monitoring and assistance, and conclude with directions for continued research.

The need for a robust anomaly detection model is as essential as a prediction model for any intelligent smart home to function in a dynamic world. For a smart environment to perform anomaly detection, it should be capable of applying the limited experience of environmental event history to a rapidly changing environment, where event occurrences are related by temporal relations. For example, if we are monitoring the well being of an individual in a smart home and the individual has not opened the refrigerator all day as they normally do, this should be reported to the individual and the caregiver. Similarly, if the resident turned on the bathwater, but has not turned it off before

going to bed, the resident or the caregiver should be notified, and the smart home could possibly intervene by turning off the water.

2 Methods

Temporal Intervals. Allen listed thirteen relations (visualized in Fig. 1) comprising a temporal logic: before, after, meets, meet-by, overlaps, overlapped-by, starts, started-by, finishes, finished-by, during, contains, and equals [7]. These temporal relations play a major role in identifying temporal activities which occur in a smart home. Consider, for instance, a case where the inhabitant turns the Television on before sitting on the couch. We notice that these two activities, turning on the TV and sitting on the couch, are frequently related in time according to the “before” temporal relation. Therefore, when the relationship is violated, an anomaly is noted.

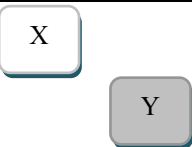
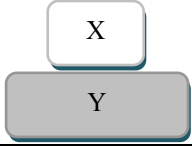
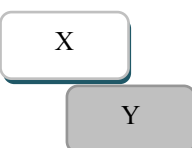
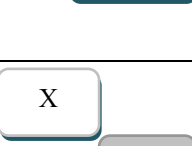
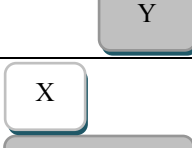
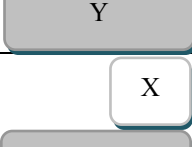
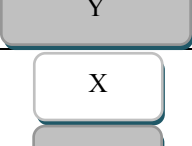
Temporal relations	Visualization	Interval constraints
X Before Y Y After X		Start(X) < Start(Y); End(X) < Start(Y)
X During Y Y Contains X		Start(X) > Start(Y); End(X) < End(Y)
X Overlaps Y Y Overlapped-by X		Start(X) < Start(Y); Start(Y) < End(X); End(X) < End(Y)
X Meets Y Y Met-by X		Start(Y) = End(X)
X Starts Y Y Started-by X		Start(X) = Start(Y); End(X) ≠ End(Y)
X Finishes Y Y Finished-by X		Start(X) ≠ Start(Y); End(X) = End(Y)
X Equals Y		Start(X) = Start(Y); End(X) = End(Y)

Figure 1. Temporal relations representation.

Anomaly detection is most accurate when it is based on behaviours that are frequent and predictable. As a result, we look for temporal interactions only among the most frequent activities that are observed in resident behaviour. This filtering step also greatly reduces the computational cost of the algorithm. To accomplish this task, we mine the data for frequent sequential patterns using the Apriori algorithm [8]. The input to the algorithm is a file of raw sensor events, each tagged with a date and time, and the result is a list of frequently-occurring sequences of events. Next, we identify temporal relations that occurred between events in these frequent sequences. The final step involves calculating the probability of a given event occurring (or not occurring), which forms the basis for anomaly detection.

The temporal relations that are useful for anomaly detection are the before, contains, overlaps, meets, starts, started-by, finishes, finished-by, and equals relations. Because we want to detect an anomaly as it occurs (and not after the fact), the remaining temporal relations - after, during, overlapped-by, and met-by – are not included in our anomaly detection process.

Let us focus now on how to calculate the probability that event Z will occur (in this case, the start of the event interval). Evidence for this probability is based on the occurrence of other events that have a temporal relationship with Z, and is accumulated over all such related events. First consider the probability of Z occurring given that the start of the temporal interval for event Y has been detected. The formula to calculate the probability of event Z based on the occurrence of event Y and its temporal relationship with Z is given by Equation (1). Note that the equation is based on the observed frequency of the observed temporal relationships between Y and Z as well as the number of occurrences of Y in the collected event history.

$$P(Z|Y) = \frac{| \text{Before}(Z,Y) | + | \text{Contains}(Z,Y) | + | \text{Overlaps}(Z,Y) | + | \text{Meets}(Z,Y) | + | \text{Starts}(Z,Y) | + | \text{StartedBy}(Z,Y) | + | \text{Finishes}(Z,Y) | + | \text{FinishedBy}(Z,Y) | + | \text{Equals}(Z,Y) |}{|Y|} \quad (1)$$

The previous discussion showed how to calculate the likelihood of event Z given the occurrence of one other event Y. Now consider the case where we want to combine evidence from multiple events that have a temporal relationship with Z. In our example we have observed the start of event X and the start of event Y, and want to establish the likelihood of event Z occurring. The combined probability is computed as:

$$\begin{aligned} P(Z|X \cup Y) &= P(Z \cap (X \cup Y)) / P(X \cup Y) \\ &= P(Z \cap X) \cup P(Z \cap Y) / P(X) + P(Y) - P(X \cap Y) \\ &= P(Z|X).P(X) + P(Z|Y)P(Y) / P(X) + P(Y) - P(X \cap Y) \end{aligned} \quad (2)$$

Using Equation 2 we can calculate the likelihood of event Z occurring based on every event we have observed on a given day to that point in time. We can also calculate the likelihood that an event Z *does not* occur as $P(\neg Z) = 1 - P(Z)$, the inverse of the probability that event Z does occur. Finally, we calculate the anomaly value of event Z using Equation 3.

$$\text{Anomaly}_Z = 1 - P(Z) \quad (3)$$

Notice that if the event has a probability approaching 1 and has occurred, this is not considered an anomaly. On the other hand, if the probability of the event we just observed is close to 0, then this is an unusual event and should be considered an anomaly. The point at which these anomalies are considered surprising enough to be reported is based somewhat on the data itself [9]. If the probability of an event is based on the occurrence of other events which themselves rarely occur, then the evidence supporting the occurrence of the event is not as strong. In this case, if the event has a low probability yet does occur, it should be considered less anomalous than if the supporting evidence itself appears with great frequency. Consistent with this theory, we calculate the mean

and standard deviation of event frequencies over the set of events in the resident's action history. Events are reported as anomalies (or, conversely, the absence of an event) if it does occur and its anomaly value is greater than the mean + 2 standard deviations, which is a common threshold for identifying outliers in data.

MavHome Smart Home. The algorithms described here are part of the MavHome multi-disciplinary project, which has been engaged in the creation of adaptive and versatile home and workplace environments in the past few years [10]. The goal of the MavHome project is to create a smart home that can act as an intelligent agent. The home perceives the state of the environment and its residents using sensors, reasons about the state and possible actions using machine learning algorithms, and act on the state using powerline controllers. In order to design a smart environment, we need to design machine learning algorithms that can identify, predict, and reason about resident behaviours. The objective of our initial MavHome study was to determine if our algorithms could learn an automation policy that would reduce the number of manual interactions the resident performed in a smart environment. Our machine learning algorithms did accurate predict resident activities and substantially reduce the average number of daily manual interactions [10].

The MavHome algorithms are tested in two physical environments. One is a smart apartment called the MavPad and another is a smart workplace environment, the MavLab. Our experiments are based on two months of real activity data collected in the MavLab working environment. During that time, a student volunteer performed his normal daily work activities in this environment. All interactions with lights, blinds, fans, and electronic devices were performed using X10 controllers, so that all sensor and interaction events could be captured in a text file. The layout of sensors and controllers in the MavLab is shown in Fig. 2. The data collection system consists of an array of sensors and X10 powerline controllers, connected using an in-house sensor network. As shown in Fig. 2, MavLab consists of a presentation area, a kitchen, student desks, a lounge, and a faculty room. There are over 100 sensors deployed in the MavLab that include light, temperature, humidity, and reed switches.

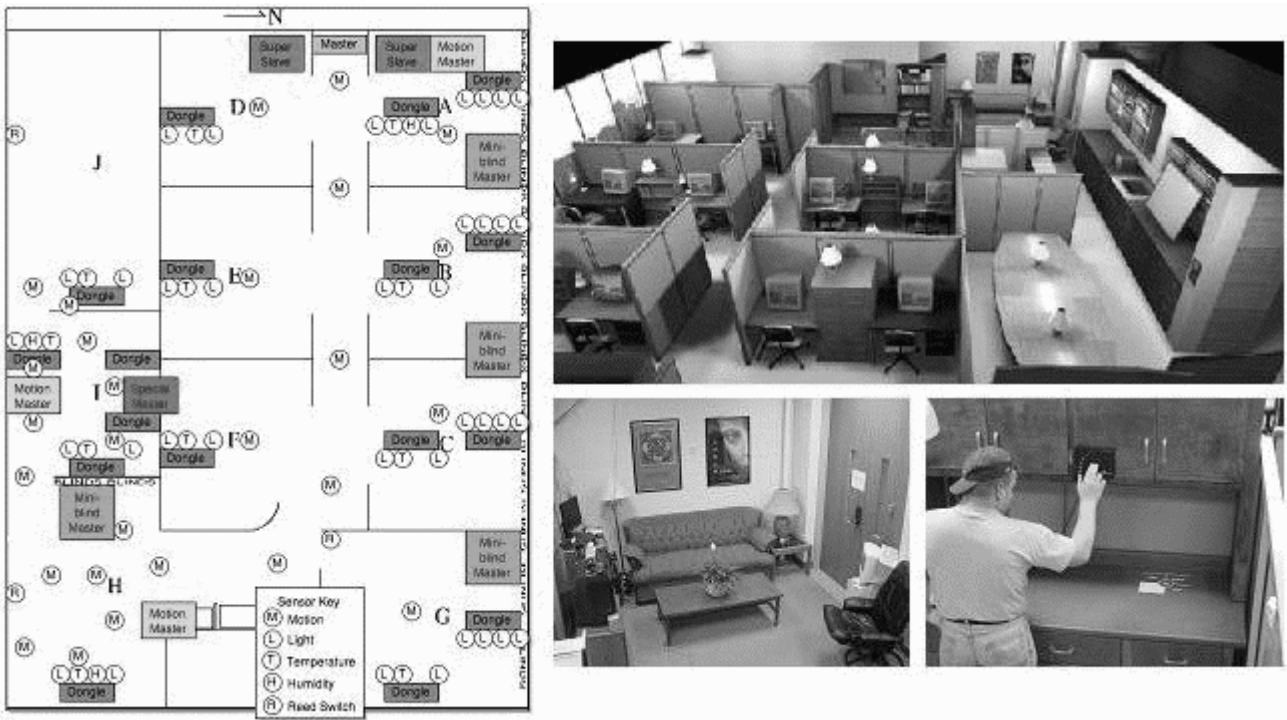


Figure 2. MavLab Argus Sensor Network (M-Motion sensor, L-Light sensor, T-Temperature sensor, H-Humidity sensor, R- Reed switch sensor, S-Smoke sensor, C- Gas sensor).

In addition, we created a synthetic data generator to validate our approach. The data generator allows us to input event sequences corresponding to frequent activities, and specify when the sequences occur. Randomness is incorporated into the time at which the events occur within a sequence using a Gaussian distribution. We developed a model of a user’s pattern which consists of a number of different activities involving three rooms in an environment and eight devices. Our synthetic data set contains about 4,000 actions representing two months of activities.

3 Results

We validate our algorithm by applying it to our real and synthetic datasets. We train the model based on 59 days of data and test the model on one day of activities. We use the training set to form the frequent item sets and identify temporal relations shared between them. The temporal relations formed in these data sets show some interesting patterns and indicate relations that are of interest. Table 1 summarizes characteristics of the datasets we used for the experiments.

Next, we perform frequent itemset mining and identify the most frequent activities in the training dataset. Then we read these temporal relations into our anomaly detection tool which calculates evidence for each possible event and outputs anomalies that are detected in the test set data. After manually inspecting the data, we report the number of true and false anomalies that are reported. Tables 2 and 3 display results from the synthetic and real datasets, respectively. Because anomalies are detected in real time as events are observed, we list anomalies in the order they are detected.

Based on a manual inspection of the data we see that the anomaly detection algorithm performed well on synthetic data – all of the expected anomalies were detected and no false positives were reported. In the real data no anomalies are reported. This is consistent with the nature of the data which does not contain anomalous events, and reflects the fact the anomalies should be, and are in fact, rare. We see that the approach is robust and does not report false anomalies in this case. The graph in Fig. 3 visualizes the anomaly values for frequent events in the synthetic and real datasets. We notice that the spikes visible in the synthetic datasets are clear indication of anomaly, which is consistent with our expectation for the outcome of this experiment.

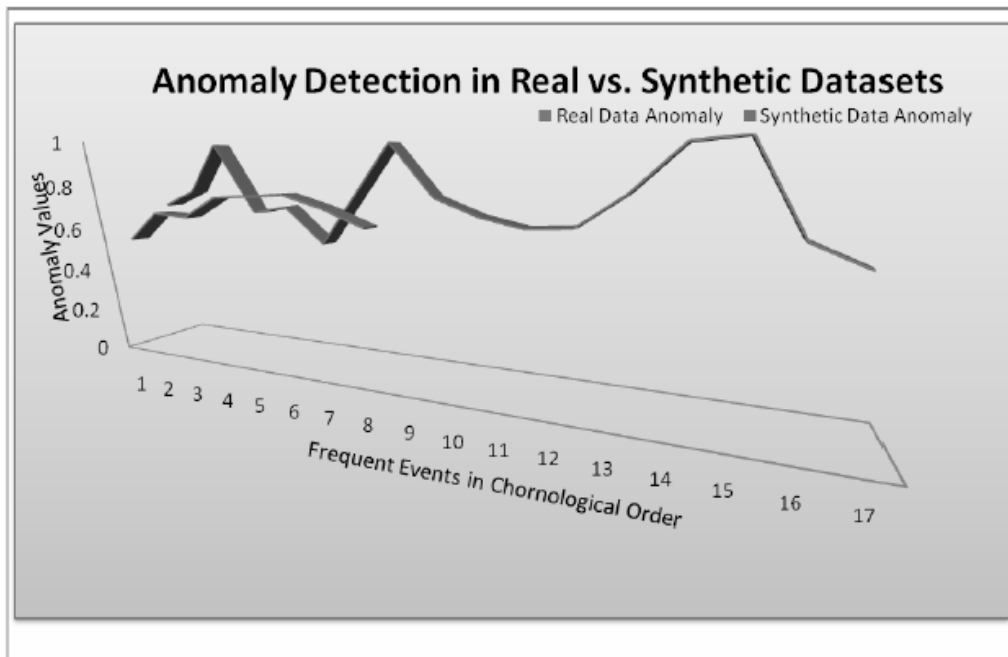


Figure 3. Anomaly detection on test sets of real and synthetic data in 3-D. The anomaly value is plotted for each possible activity as actual events are observed.

4 Conclusions

The experimental results on synthetic data provide evidence that our algorithm is capable of identifying anomalous events based on temporal relationship information. The results applied to real data brought insights to the activities that were being performed in the MavLab setting. In both cases these types of surprising behaviours should be reported to the resident and possibly their caregiver. The caregiver could respond according to the health-critical nature of the anomaly and any additional information they may have available.

A future use of anomaly detection is its use for reminder assistance. If the resident queries the algorithm for the next routine activity, the activity or activities with the greatest probability will be provided. Similarly, if an anomaly is detected, the smart environment can first initiate contact with the resident and provide a reminder of the activity that is usually performed at that time. Autominder [11] is an example of an existing reminder system. In contrast to our approach which learns a reminder schedule from observed events, Autominder's reminder schedule is preprogrammed. Autominder uses techniques such as dynamic programming and Bayesian learning to dynamically adjust this schedule in a way that accommodates dynamic changes in a person's daily routines.

Temporal reasoning enhances data mining in smart environments by adding information about expected temporal interactions between resident activities. Based on our study, we conclude that the use of temporal relations provides us with an effective new approach for anomaly detection. We tested our algorithm on relatively small datasets, but will next target larger datasets with real activity data collected over a six month time span. Other future directions of this work also include improving activity prediction using temporal relations in smart home data. One challenge this work introduces is determining which observed events belong to the same activity, and thus the same temporal interval. In this study we grouped events that turned a device on together with those that turned the same device off. However, for a more extensive study we need to determine a general method for grouping events.

Acknowledgements

This work is supported by NSF grant IIS-0121297.

References

1. Weisman J. Aging population poses global challenges. *The Washington Post*. 2005 Feb 2; Sect. A:1.
2. Heierman EO, Cook DJ. Improving home automation by discovering regularly occurring device usage patterns. In: *Proceedings of the Third IEEE International Conference on Data Mining*; 2003 Dec 19-22; Melbourne, Florida. IEEE Computer Society; 2003. p. 537-540.
3. Liao L, Fox D, Kautz H. Location-based activity recognition using relational Markov networks. In: *Proceedings of the International Joint Conference on Artificial Intelligence*; 2005 Jul 30 – Aug 5; Edinburgh, Scotland. Professional Book Center; 2005. p. 773-778.
4. Lühr S, Venkatesh S, West GAW, Bu HH. Explicit State Duration HMM for Abnormality Detection in Sequences of Human Activity. In: *PRICAI 2004: Trends in Artificial Intelligence, Eighth Pacific Rim International Conference on Artificial Intelligence*; 2004 Aug 9-13; Auckland, New Zealand. Springer; 2004. p. 983-984.
5. Gopalratnam K, Cook DJ. Online sequential prediction via incremental parsing: The Active LeZi algorithm. *IEEE Intelligent Systems*. 2007; 22(1):1-8.
6. Gottfried B, Guesgen HW, Hübner S. Spatiotemporal Reasoning for Smart Homes. In: Augusto JC, Nugent CD, editors. *Designing smart homes*. Springer Verlag; 2006. p. 16-34.

7. Allen JF, Ferguson G. Actions and events in interval temporal logic. *Journal of Logic and Computation*. 1994; 4(5):531-579.
8. Agrawal R, Srikant R. Mining sequential patterns. In: Yu PS, Chen ALP. *Proceedings of the Eleventh International Conference on Data Engineering*; 1995 Mar 6-10; Taipei, Taiwan. IEEE Computer Society; 1995. p. 3-14.
9. Noble CC, Cook DJ. Graph-based anomaly detection. In: Getoor L, Senator TE, Domingos P, Faloutsos C, editors. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; 2003 Aug 24-27; Washington, D.C. ACM; 2003. p. 631-636.
10. Youngblood GM, Cook DJ. Data mining for hierarchical model creation. *IEEE Transactions on Systems, Man, and Cybernetics*. 2007; 37(4):1-12.
11. Pollack ME. Intelligent Technology for an Aging Population: The use of AI to assist elders with cognitive impairment. *AI Magazine summer issue*. 2005; 26(2): 9-24.

Table 1. Characteristics of the synthetic and real datasets.

Datasets	#Days	#Possible events	#Frequent intervals	Size
Synthetic (train)	59	8	1703	105KB
Real (train)	59	17	1523	103KB
Synthetic (test)	1	8	17	2KB
Real (test)	1	17	9	1KB

Table 2. Anomaly detection in the test set for the synthetic dataset

Event Order	Frequent Event	Evidence	Anomaly	Reported
1	Lamp	0.30	0.70	No
2	Lamp	0.23	0.77	No
3	Lamp	0.01	0.99	Yes
4	Fan	0.32	0.68	No
5	Cooker	0.29	0.71	No
6	Lamp	0.45	0.55	No
7	Lamp	0.23	0.77	No
8	Lamp	0.01	0.99	Yes
9	Lamp	0.23	0.77	No
10	Fan	0.30	0.70	No
11	Cooker	0.34	0.66	No
12	Lamp	0.33	0.67	No
13	Lamp	0.20	0.80	No
14	Lamp	0.02	0.98	No
15	Lamp	0.00	1.00	Yes
16	Fan	0.34	0.66	No
17	Cooker	0.42	0.58	No
Anomaly Cut-off Threshold ($\mu + 2\sigma$)				0.99

Table 3. Anomaly detection in the real dataset.

Event Order	Frequent Event	Evidence	Anomaly	Reported
1	J10	0.45	0.55	No
2	J11	0.32	0.68	No
3	A11	0.33	0.67	No
4	A15	0.24	0.76	No
5	A11	0.23	0.77	No
6	A15	0.22	0.78	No
7	I11	0.27	0.73	No
8	I14	0.34	0.66	No
Anomaly Cut-off Threshold ($\mu + 2\sigma$)				0.84