Improving Home Automation by Discovering Regularly Occurring Device Usage Patterns

Edwin O. Heierman, III Diane J. Cook
Department of Computer Science and Engineering
University of Texas at Arlington
{heierman,cook}@cse.uta.edu

Abstract

The data stream captured by recording inhabitant-device interactions in an environment can be mined to discover significant patterns, which an intelligent agent could use to automate device interactions. However, this knowledge discovery problem is complicated by several challenges, such as excessive noise in the data, data that does not naturally exist as transactions, a need to operate in real time, and a domain where frequency may not be the best discriminator. In this paper, we propose a novel data mining technique that addresses these challenges and discovers regularly-occurring interactions with a smart home. We also discuss a case study that shows the data mining technique can improve the accuracy of two prediction algorithms, thus demonstrating multiple uses for a home automation system. Finally, we present an analysis of the algorithm and results obtained using inhabitant interactions.

1. Introduction

Several research efforts are focused on home automation. The Intelligent Room [2] uses an array of sensors and AI techniques to interpret inhabitant activities within the environment and provide automated assistance. The Neural Network House [6] balances the goals of anticipating inhabitant needs and energy conservation by use of a neural network. The MavHome [3] project uses an intelligent and versatile home agent that perceives the state of the home through sensors and acts on the environment through effectors.

Our research is unique in that it looks at improving home automation by discovering regular usage patterns in a data stream collected from user interactions with the home appliances and devices. These patterns, if predictable, can be automated for the user or even improved in a way that achieves the same end result in less time or energy cost. However, the numerous noisy patterns contained within the data complicate the problem. For example, it would be difficult, and undesirable, to automate appliance interactions for random and frequent trips to the kitchen to get a drink of water. It is important to filter out this noise, because embedded within the collected data are significant patterns that can be automated, such as the appliance interactions that occur when inhabitants wake and prepare to leave for work. Our goal is to discover significant patterns worthy of automation so that an agent can reduce inhabitant interactions, without generating unnecessary device actions.

Therefore, we propose a framework based on a novel data-mining algorithm ED (**E**pisode **D**iscovery) that discovers behavior patterns within a sequential data stream. The pattern knowledge and corresponding filtered data can be provided to a prediction algorithm of decision learner to improve home automation. In addition, the framework processes the interactions incrementally and can be used in a real-time system.

Several important characteristics of the home automation problem influence our discovery technique. First, the input sequence consists of histories of device interactions (episodes) with no indicated pattern start or stop points. Second, ordering information of actions within the pattern must be discovered. Third, the discovery needs to balance episode length with frequency and periodicity or regularity. Finally, the dataset will be of moderate size, which allows for the use of techniques that might not be suitable for extremely large datasets.

With these characteristics in mind, we developed an approach that mines a device activity stream to discover subsequences, or episodes, that are closely related in time. These episodes may be totally or partially ordered, and are evaluated based on information theory principles.

The following scenario, executed by the MavHome smart home for an inhabitant named Bob, illustrates the potential uses of the ED algorithm. Bob's alarm goes off at 7:00, which signals MavHome to turn on the bedroom light as well as the coffee maker in the kitchen. When Bob steps into the bathroom, MavHome turns on the light, displays the morning news on the bathroom video screen, and turns on the shower. When Bob finishes grooming, the bathroom light turns off while the kitchen light and menu/schedule display turns on, and the news program moves to the kitchen screen. When Bob leaves for work, MavHome secures the home, and starts the

lawn sprinklers. Because the refrigerator is low on milk and cheese, MavHome places a grocery order to arrive just before Bob comes home.

In order to provide such a level of automation, MavHome can use ED to mine the event history of the devices. Upon mining the data, ED discovers that some activities such as {alarm on, alarm off, bedroom light on, coffee maker on, bathroom light on, bathroom video on, shower on} occur on a daily basis, while others such as {sprinkler on, sprinkler off} occur weekly. ED identifies these collections as significant episodes because they provide compression of the event history. Once the significant episodes are identified, MavHome can use this knowledge to provide the desired device automation.

Several works address the problem of discovering sequences. Agrawal [1] mines sequential patterns from time-ordered transactions using variations of the Apriori property. Mannilla [5] discovers frequent parallel and serial episodes using a window of user-defined size to slide over an event stream. Our work uses the same definition for an episode, and employs the sliding window to identify episodes. The Apriori support measure, however, is supplanted with a Minimum Description Length (MDL) evaluation measure.

2. The Episode Discovery (ED) Algorithm

Given an input stream *S* of event occurrences *O*, ED:

1. Partitions *S* into Maximal Episodes, P_{max}.
2. Creates Itemsets, *I*, from the Maximal Episodes.
3. Creates a Candidate Significant Episode, *C*, for each Itemset *I*, and computes one or more Significance Values, *V*, for each Candidate.
4. Identifies Significant Episodes by evaluating the Significance Values of the candidates.

First, our technique generates maximal episodes, P_i, from the input sequence, *S*, by incrementally processing each event occurrence, O_j. An episode window maintains the occurrences and is pruned when an occurrence is outside of the allowable window time frame due to the addition of a new event occurrence. The window contents prior to pruning are maximal for that particular window instance, and are used to generate a maximal episode.

Next, the algorithm constructs an initial collection of itemsets, one for each maximal episode. Additional itemsets are generated so that the episode subsets of the maximal episodes can be evaluated for significance. To avoid generating the power set of each maximal episode itemset, ED must prune the complete set of potential itemsets in a tractable manner, while ensuring itemsets leading to significant episodes are retained.

Because frequency is not our only discriminator of interestingness, pruning based on the Apriori measure of frequency is not sufficient. Nevertheless, we can prune the itemset search space by selecting a subset of an episode itemset as an additional itemset based on one of the following conditions:

- The subset represents the intersection of multiple maximal episode itemsets. Because the subset represents event occurrences in multiple episodes, it may be more significant than its parents.
- The subset represents the difference between a maximal episode itemset and one of its subset itemsets. If a subset itemset is significant, then the remainder of the maximal episode itemset must be evaluated to see if it is also significant.

Our approach relies on the following principle to prune the itemset space: subset itemsets that have the same episode occurrences as their parent itemset do not need to be generated as candidates. Because these subset itemsets are shorter in length, but have the same frequency and regularity as their parent, their value cannot be greater than the parent. Our pruning method generates smaller itemsets from larger ones, which is essentially the opposite of an Apriori approach [1] where larger itemsets are generated from smaller ones.

Once the itemsets have been generated, a significant episode candidate C_n is created for each itemset I_n. Next, each maximal episode is compared with the I_n of each candidate. If I_n contains all or part of the maximal episode, then the maximal episode is added to the episode set of the candidate as an occurrence of the itemset.

ED evaluates the candidates by making use of the MDL principle, which targets patterns that can be used to minimize the description length of a database by replacing each instance of the pattern with a pointer to the pattern definition. Our MDL-based evaluation measure thus identifies patterns that balance frequency and length. Instances of a regular (daily, weekly) sequence can be removed without storing a pointer to the sequence definition, because the regularity of the sequence is stored with the sequence definition. Regular sequences thus further compress the description length of the data. The larger the potential amount of compression a pattern provides, the greater the impact that results from automating the pattern. Therefore, the resulting amount of compression represents the pattern's significance value. Currently, we detect daily and weekly periodic patterns, and adjust the description length to account for too frequent or infrequent deviations from the expected regularity.

Once the candidates have been evaluated, ED greedily identifies the significant episodes as those meeting a user-configurable minimum significance (compression) value. Because a high level of confidence is desired before automating an episode, we chose 80% as the minimum compression. After selecting a candidate, the algorithm marks the events that represent instances of

the candidate's pattern, which allows ED to construct a filtered input stream. These steps are repeated until all candidates have been processed.

The knowledge that ED obtains by mining the device-interaction history can be used in a variety of ways. The filtered input stream could be provided to a prediction algorithm, improving the algorithm's accuracy by removing noise from its training data. Also, the significant episodes can be used to segment a decision learner's state space into meaningful partitions and enhance scalability of the algorithm. Finally, an episode's temporal information, such as episode start and stop times, can be used to enhance automation decisions.

3. Case Study

To validate the ED algorithm, we conducted a case study to demonstrate that regular episodes could be discovered, and also that the filtered input stream, significant episodes, and temporal knowledge of the significant episodes could be used effectively. We selected the **I**ncremental **P**robabilistic **A**ction **M**odeling (IPAM) sequential prediction algorithm [4] and a **B**ack-**P**ropagation **N**eural **N**etwork (BPNN) for this case study. IPAM is a frequency-based prediction algorithm that maintains a probability distribution for the next event given the current state. Both algorithms could be used to predict the next action that might occur in a home environment. Because these algorithms by themselves may not fare well in the home environment due to noisy data, we felt they would be ideal candidates. The goal was to show that by training each algorithm on filtered data and using the significant episode temporal knowledge when performing a prediction, ED could improve the predictive accuracy of the algorithms.

Using a synthetic data generator, we created five randomly-generated scenarios from a device usage description that covers typical device interactions over a six-month time period. The scenario generator supports generating random noisy patterns, allows the occurrence of the patterns to overlap, and allows the pattern order to be varied. Our scenario definition includes twenty-two devices with on and off states. Fourteen regularly-occurring daily and weekly patterns were defined that ED should discover, as well as sixty-eight noisy patterns. Eight devices were used as part of both the regularly-occurring patterns and the noisy patterns. Each scenario contains close to 13,000 device interactions, of which less than 5,000 are part of a regularly-occurring pattern.

We tested the algorithms in the following manner. First we trained each prediction algorithm (IPAM, BPNN) on the entire scenario dataset, using parameter values that maximized the performance based on initial testing. Next, ED mined the scenario dataset to identify the significant episodes, and marked the device interactions that were part of the discovered significant episodes. ED was configured to use a fifteen-minute sliding window in order to partition the input stream. We then trained a version of each algorithm (IPAM+ED, BPNN+ED) on the filtered data.

Once the algorithms were trained, the synthetic data generator used the same scenario description to randomly create a test scenario that covered the same twenty-two devices over an additional one-month period. Using the test dataset, each algorithm predicted the next event occurrence based on the current event occurrence. For IPAM+ED and BPNN+ED, we used the significant episode temporal information to perform a prediction only when the following conditions were met:

- Event *e* of the current event occurrence *O* is a member of one of the significant episodes, and
- The time of *O* falls within the time-of-day range of the episodes the significant episode represents.

Thus, the combined approach should automate regular device interactions while ignoring unnecessary interactions, which, as we noted previously, is very important for a home environment.

ED was able to correctly discover the pre-defined significant episodes in all of the datasets, and appreciably improved the accuracy of both algorithms across all five scenarios, as can be seen in Table 1. Therefore, we conclude that the knowledge discovered by ED can be used to consistently improve the predictive accuracy of the case study algorithms. One reason for the improvement was that the algorithms were trained on significant, noise-free data. In addition, the predictions were filtered so that predictions were not performed on data that was considered noise. ED not only improved the accuracy of the algorithms, it also significantly reduced the total number of false predictions (and resulting wasteful automations) each algorithm made.

Table 1 Averaged Scenario Prediction Results

Approach:	IPAM	IPAM+ED	BPNN	BPNN+ED
Accuracy:	41.0 %	73.6 %	63.6%	85.6%

4. Algorithm Analysis

We performed additional testing that compared the algorithm against a frequency-based approach, evaluated the runtime performance of the algorithm, and tested ED using real inhabitant interactions collected from our MavHome environment.

For comparison purposes, we modified ED to discover the most frequently occurring episodes. The algorithm selected an episode as significant if it occurred a minimum number of times. We varied this support level

from 50 to 200 in increments of 50, and tested the accuracy of the BPNN+ED predictor on the synthetic datasets. The results are shown in Table 2. The highest accuracy obtained was 64%, which is well below the 85.6% achieved by our MDL approach. In addition, no weekly episodes were discovered, and noisy patterns were selected as significant episodes in all of the test runs. We conclude that the ED algorithm using our MDL-based approach outperforms a frequency-based approach on the synthetic datasets.

Table 2 Frequency-based Results

Support:	50	100	150	200
Accuracy:	46%	53%	64%	54%

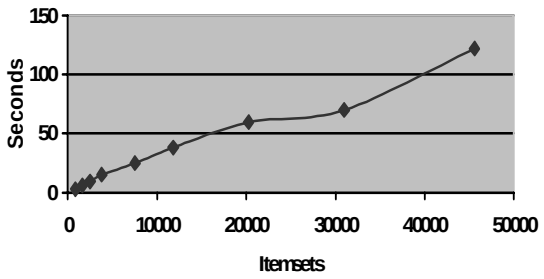

Figure 1 Itemset vs. Processing Time Plot

Our tests also show that ED is an efficient algorithm for the home automation domain. We empirically analyzed the runtime complexity of the algorithm by varying the window width, and in Figure 1 we show a plot of the number of itemsets generated versus the processing time. The results indicate that the algorithm achieves a near-linear performance on the synthetic datasets.

Finally, we collected one month of device interaction data from six participants in our MavHome environment laboratory. The dataset consists of 618 interactions contained in patterns occurring once a week, multiple times a week, and randomly. We used ED to mine the data with window widths ranging from 2 to 10 minutes and compression minimums ranging from 20% to 60%. ED was able to correctly identify the patterns of three of the inhabitants as significant episodes that occur weekly. The results of this testing highlighted the following:

- The algorithm must be improved to discover the regularity of the patterns. ED only discovered daily or weekly patterns, and was unable to discover the patterns that occurred multiple times in a week.
- The approach used to determine membership in a significant episode needs improvement. Several of the significant episodes contained anomalous episode occurrences, which incorrectly increased the time-of-day range of the episodes to an excessively large value.

5. Conclusions and Future Work

We have demonstrated that ED provides an efficient mechanism for identifying significant episodes in sequential data. ED can be used to comprehend the nature of the activities occurring within the environment, and to improve the predictive accuracy of a home agent. We anticipate that ED will serve as a penultimate component that helps identify the activities that a smart home agent should automate.

In our future work, we intend to test ED on interaction data collected from the MavHome residence, and to incorporate additional prediction and decision learner algorithms. In addition, we are currently applying auto-correlation techniques to automatically detect regularity intervals in observe data. We are also investigating automatically discovering the optimal window width by evaluating compression values between windows. Finally, we intend to improve the information provided when identifying event occurrences that may be members of a significant episode.

6. REFERENCES

[1] R. Agrawal and R. Srikant, "Mining sequential patterns," *Proc. 11th International Conference Data Engineering (ICDE 1995)*, Taipei, Taiwan, pp. 3-14, March 1995.

[2] M. Coen. Design principles for Intelligent Environments. *AAAI Spring Symposium*, Stanford, pp. 36-43, March 1998.

[3] S. Das, D. Cook, A. Bhattacharaya, E. Heierman, and T. Lin. The Role of Prediction Algorithms in the MavHome Smart Home Architecture. *IEEE Wireless Communications*, vol. 9, no. 6, pp. 77-84, December 2002.

[4] B. Davison and H. Hirsh. Predicting Sequences of User Actions. In *Technical Report*, Rutgers, The State University of New York, 1998.

[5] H. Mannila, H. Toivonen, and A. Verkamo, "Discovering frequent episodes in sequences," *Proc. 1st International Conference on Knowledge Discovery and Data Mining (KDD'95)*, Montreal, Canada, pp. 210-215, August 1995.

[6] M. Mozer. An intelligent environment must be adaptive. *IEEE Intelligent Systems*, vol. 14, no. 2, pp. 11-13, March/April 1999.