



International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

www.iasir.net

Various Techniques for Activity Recognition: An Overview

Amar Jyoti Kumar¹, Nikumani Choudhury², Moumita Roy³

Department of Information Technology
IST, Gauhati University, Guwahati, India

Abstract: The Information Technology today has influenced every aspect of human life and its newest impact has been in the field of Smart Home Environment for providing better proactive services to its inhabitants. In such a scenario, predicting the next event that is to occur is a very important factor wherein comes the play of our Activity Recognition. In this paper, different types of technique for Activity Recognition have been studied and the advances of work in a particular technique have been discussed and the advances made by researchers in making the activity recognition algorithm better has also been discussed.

Keywords: Minimum Description Length (MDL), PPM, Markov Model

I. Introduction

Activity recognition is a process of recognizing the actions and goals of one or more agents from the observations of the agents' actions and its environmental conditions. Activity recognition tries to detect various physical activities of the user such as walking, sitting, driving, running or standing still. Since the 1980s, this field of research has been a major research topic because of its potential to provide personalized support in various applications and its connection in various fields of study such as medicine, human computer interaction such as Smart environment or GSM, or sociology, etc.[1] To understand activity recognition better, we consider the following scenario. An elderly man who stays alone in his apartment wakes up in the morning. He turns the stove on to make a cup of tea, then switches on the toaster oven, and then takes some bread and jam from his cupboard. After he takes his morning medication, then a computer-generated voice reminds him to turn the toaster off. Later that day, his daughter from a secure website accesses and scans the check-list that was created by the sensor network in her father's room. She finds out that her father is eating normally, taking his medicine timely, and also managing his daily life quite well on his own. This information relaxes her mind that her father is ok as he is taking his food and medicine timely.

There have been different approaches to Activity Recognition [1]. Some of them are:

- ☐ Activity recognition through logic and reasoning
- ☐ Activity recognition through probabilistic reasoning
- ☐ Wi-Fi-based activity recognition
- ☐ Data mining based approach to activity recognition

II. Some of the Activity Recognition Algorithms

A. Activity Recognition using SHIP:

The Smart Home Inhabitant Prediction algorithm works on the principle of sequence matching. It matches the most recent event sequences with sequences in collected histories. Whenever a command is issued by an inhabitant to a device, it gets recorded in the inhabitant action history. A match identifies a sequence in the event history that matches with the immediate event history. For this a match queue is maintained. The SHIP algorithm consists of the following two steps. First, whenever a new action is recorded, the match queue will get updated. At a time t in a state s we compute $l_t(s, a)$, as the length of longest sequences ending with action a in state s and match the history sequence immediately prior to time t . In addition, we define a frequency measure $f(s, a)$ which represents the number of times the action a has been taken from the current state. In the second step, SHIP evaluates the matches based on a combination of match length and frequency, according to the equation:

$$R_t(s, a) = \alpha \frac{l_t(s, a)}{\sum_t l_t(s, a_i)} + (1 - \alpha) \frac{f(s, a)}{\sum_t f(s, a_i)}$$

SHIP predicts the action corresponding to the greatest $R_t(s, a)$ value obtained[2]. The main advantage of this algorithm is its design simplicity and ability to incrementally use all its collected history. But the disadvantage is that the entire action history has to be stored as well as processed off line, which is quixotic for large prediction tasks over a long duration.

B. Activity Recognition using Active Lezi (ALZ):

The Active Lezi algorithm uses principles of information theory for processing action sequences. It utilizes a

sequential prediction algorithm because device-inhabitant interactions are characterized as Markov chain of events. As ALZ uses the LZ78 data compression technique which happens to be an incremental parsing algorithm, so the ALZ is also an online algorithm.

The LZ78 algorithm parses incrementally an input string $s_1, s_2, s_3, \dots, s_i$ into $c(i)$ substrings $w_1, w_2, w_{c(i)}$ such that for all $j > 0$, the prefix of substring w_j is equal to some w_i for $1 < i < j$. This algorithm maintains a record of all contexts of each phrase. This information is then used for compressing and reconstructing text strings in an online manner.[2]

Active LeZi enhances the original LZ78 algorithm. It recaptures information which would be lost across phrase boundaries. And for this, it maintains a variable-length window for the processed symbols, the length being equal to the length of the longest phrase that has been obtained so far. Based on this information, statistics are gathered on all possible contexts seen.

For prediction, ALZ calculates the probability of every action that occurs in the parsed sequence, and predicts that action which has the highest probability. To enhance the prediction, ALZ uses ideas from the Prediction by Partial Match (PPM) family of predictors. This approach has been used by Bhattacharya, et al. [3] in a predictive framework based on LZ78[2]. PPM algorithms consider Markov models of different order to build a probability distribution by weighing the different order models appropriately. ALZ starts by building an order- k Markov model, which then uses the PPM mechanism for gathering information from all lower – order models. This information is used for determining the probability value of the next symbol

The principal advantage of this algorithm is its ability to use information from different order models and also to process information in an incremental manner. But this algorithm in its prediction does not incorporate actual event times. [4]

C. Activity Recognition using Episode Discovery (ED):

Episode Discovery is a prediction algorithm wherein it identifies significant episodes from the action history of its inhabitant. A significant episode can be said to be a collection of device events which maybe either ordered, unordered or partially ordered.

The goal of this algorithm is to mine the input sequence of actions in order to obtain the significant episodes. Once the significant episodes are mined, actions can then be automated based upon how important the discovered pattern is as well as the predictive accuracy of the next event. An input stream O processed by ED consists of a time ordered sequence of events $\{s, t\}$, where s is a symbol representing an interaction and t is a time stamp. Given an input stream O , ED algorithm is given as:

1. Partition O into Maximal Episodes E_{max} .
2. Create Candidate Items C from the Maximal Episodes.
3. Create Additional Candidates.
4. Compute a Compression Value V for each candidate.
5. Identify Interesting Episodes P by evaluating the Compression Value of the generated candidates.
6. Output Interesting Episodes.

III. Related Works

Here we provide a brief overview of the works done by various researchers, which are related to the domain Episode Discovery. A subclass of sequential pattern mining techniques specifically addresses time ordered input sequences that do not naturally contain transactions but consist of events. For this category of problems, a partitioning technique is needed to collect related events into episodes. The resulting episodes are evaluated in order to discover repeating patterns that are interesting. Mannila, Toivonen and Verkamo [6] discovered frequent episodes in an event sequence. A user-defined event window partitions the event stream into overlapping collections of events that are close to each other in time into episodes. These collections are examined to find a class of frequent episodes by use of the Apriori property. Supported classes include all serial episodes, all parallel episodes, exact episode topologies, and all partial orders.

Laxman [7] examined various techniques to make the ED algorithm faster.

Srinivasan [8] took an alternative approach by first discovering significant intervals that met a minimum interval confidence over a pre-defined periodicity. After the significant intervals were discovered, they were then used for identifying episodes by the use of an event window and an Apriori-based approach.

One limitation of the frequent episode algorithm [6] was the need to provide the event-folding window and support value parameters. Heuristics and experimentation were required for determining the proper settings. Secondly, the number of windows containing the pattern determined the frequency counts. And because the windows are overlapping, this cannot be a true statistical count of the number of occurrences of the pattern. Also, the number of windows was influenced by the window size, which led to different counts for different window sizes.

Laxman [7] proposed non-overlapping and non-interleaved counting approach as a solution to this limitation. A third limitation was the need to understand the nature of the structure of the frequent patterns to be discovered. In

some cases, it may not be known if interesting patterns are ordered or unordered. Fourthly, frequency was the only measure of interestingness whereas other characteristics such as periodicity might also be interesting. Srinivasan [8] accounted for periodicity, but that required the intervals to be predefined. Finally, no evaluation measure for the discovered episodes was provided. A set of frequent episodes produced by one window size and frequency support could not be compared with a collection produced by another set of parameter values to determine which parameter settings resulted in the best information. So he used the Hybrid Apriori Algorithm.

Edwin O. Heierman, III [9] used Rissanen's work [10] on Minimum Description Length (MDL) which searched for models and model classes with the shortest description length. Rissanen reasoned that by using an encoding of a dataset to reduce its description length, constraints were applied to the data that reduced the uncertainty about the nature of the data. The resulting "minimum-encoded" description captured the properties that provided the most likely explanation for the data. It then computed compression ratios of the events and classified the events as interesting based on these compression values.

Edwin and D.J. Cook [11] to validate the ED algorithm used its output to create a hierarchical hidden Markov model (HMM)[12][13]. This model was built from multiple passes through the dataset, that created an increasing hierarchy of abstraction. Experiments using simulated environment data has proven the ability of this approach to automatically create HHMMs from inhabitant interaction data.

Edwin and D.J. Cook [14] to validate the ED algorithm, 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. They selected the Incremental Probabilistic Action Modeling (IPAM) sequential prediction algorithm [15] and a Back-Propagation Neural Network (BPNN) for this case study.

Both algorithms could be used to predict the next action that might occur in a home environment. 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.

IV. Conclusion

In this paper various techniques for Activity Recognition has been discussed along with their advantages and disadvantages. These algorithms not only provide information for predicting various activities but also help in identifying anomalies. These technologies have come to a great deal in reducing human work. In future the use of activity recognition can be applied to a large sector including smart home, smart offices, smart hospitals, smart automobiles, etc.

V. References

- [1] http://en.wikipedia.org/wiki/Activity_recognition
- [2] D. J. Cook, M. Youngblood, and E. O. Heierman, "MavHome: An Agent-Based Smart Home," in Proceedings of the First IEEE Int'l Conf. on Pervasive Comp. and Comm. (PerCom'03), Vol.23, No.26, pp.521-524, March 2003.
- [3] A. Bhattacharya and S. Das. Lezi- update: An information-theoretic framework for personal mobility tracking in PCS networks.
- [4] Sira Rao, D. J. Cook et al., "MavHome: An agent based Smart Home".
- [5] MavHome: An Agent-Based Smart Home by Diane J. Cook, Michael Youngblood, Edwin O. Heierman, III, Karthik Gopalratnam, Sira Rao, Andrey Litvin, and Farhan Khawaja
- [6] H. Mannila, H. Toivonen, and A. Verkamo. "Discovering frequent episodes in sequences," In Proc. 1st International Conference on Knowledge Discovery and Data Mining (KDD'95), Montreal, Canada, pp. 210-215, August 1995.
- [7] S. Laxman, P. Sastry, and K. Unnikrishnan. "Fast algorithms for frequent episode discovery in event sequences," In Proc. 3rd International Workshop on Mining Temporal and Sequential Data (TDM-04), Seattle, WA, August 2004.
- [8] A. Srinivasan. Significant Interval and Episode Discovery in Time-Series Data, Masters Thesis, University of Texas at Arlington, Arlington, TX, 2003.
- [9] E. Heierman and D. J. Cook. "Improving home automation by discovering regularly occurring device usage patterns," In Proc. 3rd International Conference on Data Mining (ICDM'03), pp. 537-540, Melbourne, FL, November 2003.
- [10] J. Rissanen. Stochastic Complexity in Statistical Inquiry. World Scientific Publishing Company, 1989.
- [11] E. Heierman, G. M. Youngblood, and D. J. Cook. "Mining temporal sequences to discover interesting patterns," In Proc. 3rd Workshop on Mining Temporal and Sequential Data (TDM'04), Seattle, WA, August 2004.
- [12] Shai Fine, Yoram Singer, and Naftali Tishby. The Hierarchical Hidden Markov Model: Analysis and Applications. Machine Learning, 32(1):41– 62, 1998.
- [13] G. Theodorou, K. Rohanimanesh, and S. Mahadevan. Learning Hierarchical Partially Observable Markov Decision Processes for Robot Navigation, 2001. IEEE Conference on Robotics and Automation.
- [14] W. Hines and D. Montgomery. *Probability and Statistics in Engineering and Management Science*, John Wiley & Sons, 1980.
- [15] B. Davison and H. Hirsh. Predicting Sequences of User Actions. In Technical Report, Rutgers, The State University of New York, 1998.