Learning the Structure of Graphical Models Based on Discrete Time Series Data in the Context of Ambient Assisted Living

Sarah Fallmann[1,2]*, Johannes Kropf [2]

[1]Department of Analysis and Scientific Computing, Vienna University of Technology, Wiedner Haupstraße 8-10, 1040 Vienna, Austria;

[2]Health & Environment Department, AIT Austrian Institute of Technology GmbH, Wiener Neustadt, Austria

**sarah.fallmann.fl@ait.ac.at*

Simulation Notes Europe SNE 25(3-4), 2015, 157-164
DOI: 10.11128/sne.25.tn.10308
Received: June 16, 2015; Revised September 10, 2015;
Accepted: October 5, 2015

Abstract. This work addresses the issue of building a probabilistic system in an ambient assisted living environment to ensure a proper living for older adults. The focus lies on the early prediction of human activities based on domotic sensor data which form a temporal-sequential data set. In contrast to commonly used methods in sequential data mining, data in hidden streams and with variable temporal spans are considered. The aim in this context is to detect recurrent patterns in a stream of domestic sensor data using the Temporal-Pattern (T-Pattern) algorithm and to automatically generate probabilistic finite-state automata.

Introduction

The motivation of the work presented here is mainly driven by the demographic change and the need for ambient assisted technologies to support a longer independent living of older adults. Behavior pattern recognition based on non-obtrusive sensors is a challenging task, but is seen to be of fundamental importance for AAL technologies being accepted by the end users.[1] Human behavior can be very complex, consider, for instance, the preparation of a meal. This activity consist of many other sub activities like taking the pan out of the cupboard, taking food out of the refrigerator, switching on the stove and so on.

The current attempts to detect human behavior and activity can be classified by the type of the sensors used (1) body worn sensors, (2) video cameras and (3) domestic sensor networks. For the sake of maximal non-obtrusiveness, this work concentrates on domotic sensor networks. The data gained from domotic sensors are mostly binary and form sequential patterns. Sequential pattern mining, which discovers frequent subsequences as patterns in such a data source, is an important problem with broad applications. This includes the analysis of customer purchases, natural disasters, disease treatment, DNA sequences and others.[2]

In the literature, methods for predicting the next symbol of a sequence of strings are proposed using probabilistic suffix trees (PST) [3, 4], which usually ignore the time between subsequent events and assume no noise in terms of unknown or random events in the data. In sensor networks consisting of many sensors, or in environments with multiple persons interacting with the smart home, patterns are very often hidden in data streams and must be discovered with appropriate statistical methods. Later the probabilistic suffix tree is transformed in an automaton. The automaton cannot be learned straight forward with this kind of data, as there exists no trainings data. This is the reason why PST are used to solve this problem.

There is a demand for future smart home solutions which increase the comfort and safety of the inhabitants while saving energy at the same time. The focus lies on caring for older adults, which are independent enough to live on their own, but desire to feel more secure. The security factor comprises, among other things, the sending of help in case of an accident or warning from an open window.

This factors are treated in Ambient Assisted Living (AAL) technologies, in specific, behavior pattern recognition can handle and meet the request of older adults.[5] Nowadays state-of-the-art smart home solutions are based on simple predefined rules to control devices based on movement detectors or timers, but there is still very little intelligence built-in due to the following reasons:

- Too many systems on the market are proprietary and cannot be connected to each other
- Very few smart appliances are available on the market
- Living environments differ significantly and standardized solutions will not work
- Configuration is complex and therefore costly

One of the keys towards smart systems in the AAL context is seen to be activity recognition, and much research in recent years has focused on this topic. Although many approaches and algorithms exist, there are still many obstacles towards a system which is ready to be used in real environments in everyday life.

The aim of this work is to describe a concept for human behavior detection, which includes also the time aspect of environmental sensor events hidden in data streams. The time aspect is important, as data comes from real world settings and different sensors can send at similar timestamps. Moreover the order of the activities is important, in particular for events which are dependent on past events. The detection of high complex patterns and behaviors are important for a higher comfort of the end users.

1 Related Work

Human activity recognition is a challenging task and has been an active, fast growing and broad research area, with a multitude of approaches and algorithms proposed. Roughly, these approaches can be categorized by camera-based, environmental sensor-based, or wearable-/object-based activity recognition. This work focuses on non-obtrusive environmental sensors for activity recognition. Algorithms used in activity recognition can be divided into two major groups. The first one is based on machine learning techniques including supervised and unsupervised learning methods, the second one is based on logical modeling and reasoning.[1]

For probabilistic activity recognition a wide range of algorithms exists. This includes Hidden Markov Models (HMMs) [6], Bayes networks [7], decision trees and other classical pattern recognition methods. Among these, HMMs and Bayes networks are the most popular. These algorithms used in a supervised context are methods, which need annotated data to construct a model. On the other hand there exists unsupervised learning methods like factor analysis and mixture of Gaussians, these are learning methods without the need of preprocessing the data. These methods try to build statistical representations of the data.[8]

An approach similar to Hidden Markov Models is that of probabilistic finite state automata (PFA).[9] PFAs describe distributions over strings, but have been successfully used in several fields, including pattern recognition. Finite state machines in general are widely used for all kind of control tasks appearing in smart environments as well. The PFA is learned over a PST, where the nodes are the patterns from the T-Pattern algorithm.

PSTs are used in a wide area, for example to analyze video data [3]. Li *et al.* [3] used an action-only prediction model to represent dependencies between action units on the basis of video data. In comparison in the approach of this paper domestic sensors are used.

In the action-only prediction model, the occurrence of a sequence of events is used and compared with an empirical parameter to detect patterns. Another approach for the detection of patterns in behavior are T-Patterns [10]. T-Patterns are created for finding hidden or non-obvious temporal patterns in streams of data describing human behavior. T-Patterns are originally used for children interaction analysis [10], but find also application in detecting animal and human behavior [11].

2 Activity Recognition

The sensors used for behavior detection are off-the-shelf and low-cost conventional home automation sensors. All sensor types have been integrated in the platform and work in parallel due to the harmonization of data packets into a hardware abstraction bundle.[12]

For evaluation, three annotated datasets consisting of several weeks of data recorded in a real-world setting provided by Kasteren *et al.* [13] are used. Each dataset belongs to one house which is occupied by one person.

2.1 Algorithm

The algorithm employed relies on the work of Li *et al.* [3] and their action-only prediction model. The authors try to construct a PST out of video data. This is done in four steps. First, they find every pattern in the video data and filter out the patterns which are not frequent. The frequent patterns are broken down into individual interactions along the time line. In the last step they calculate the probabilities of the various patterns with an algorithm containing some empirical parameters.

This fits the aim of this work quite well, but it must be expanded by the important temporal aspect. This is done by following the concept in Figure 1. In contrast to the algorithm proposed in [3], which only counts the occurrences of different frequent patterns, in this work the temporal aspect of the sensor data, as well as its statistical significance, is included in the calculation of the T-Pattern algorithm and the next symbol's probability to make it more robust. Another drawback of the referenced paper are the empirical parameters, which are unpractical and a stumbling block to a fully automatic system. In the present work, this problem does not occur.

If only the significant patterns are used to calculate the probabilities, a more appropriate result can be expected.

2.2 Concept

The real environmental sequential data from the senors are preprocessed. In this step false data are deleted and with this preprocessed data the algorithm can be proceeded.

First a T-Pattern algorithm is performed with the sensor data. Followed by building a PST with the significant T-Patterns. Therefore each significant T-Pattern stands for one node and the next symbol probabilities are calculated with the Poisson distribution.

The concept how to construct the next symbol probability of states in a PST relies on the idea of maximum duration compare Figure 1. This means in every pattern-step the longest duration of the patterns in this period is used to calculate the next symbol probability for this transition.

If the maximum duration of pattern-step 11 is interesting, the calculation begins at the pattern @tt. The pattern-step 11 in this example means the transformation of @tt in @ttt, @ttk or @ttx. The duration of every suffix of the pattern @tt{t,k,x} is compared and the maximum duration is chosen to calculate the next symbol probability. All transformation probabilities, which are calculated in the way described before, are put to the correct node, where *x* describes those cases where nothing significant happens. Therefore is it ensured that the total cases in one step sums always up to one. This is important, as otherwise no automaton can be constructed out of the PST.

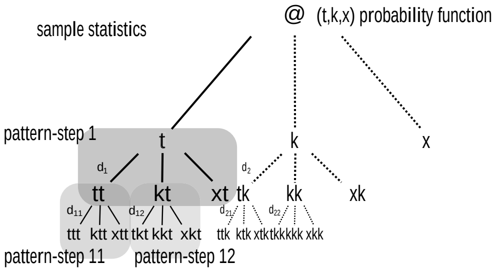

Figure 1: Concept how to calculate next symbol probability.

To complete the system, the PST must be transformed in a probabilistic suffix automaton (PSA). In [14] an algorithm can be found for this step, the result for one example can be seen in Figure 2. The PSA is created by the information of the PST in the following way. First all leaves are added to the new automaton as recurrent states and a state equal to the root of the PST. The states are connected with each other. For example if the path between root and '021' is questioned, two states are required and connected '20' and '2'.

In the next step the arcs are built, for this step the next-symbol probabilities of the PST are required. If there exists a next symbol probability after the given state, the symbol is added, and from the front symbol by symbol is removed until this state can be found in the automaton. For example '20' has a next symbol probability to '1', the constructed label is '201' after removing the first symbol '2' an already existing state '01' is found and an arc is created.

The last step is to assign state types. This is done by looking at each node and ascertaining if one of the arcs comes from a recurrent state created in step one, therefore this state also becomes a recurrent state. All other nodes are transient states. '2' is a recurrent state, as their exists an arc from the recurrent state '10' to '2'.[14]

This algorithm leads to a PSA, out of significant T-Pattern which can detect behavioral anomalies in real environments with domestic sensors.[12]

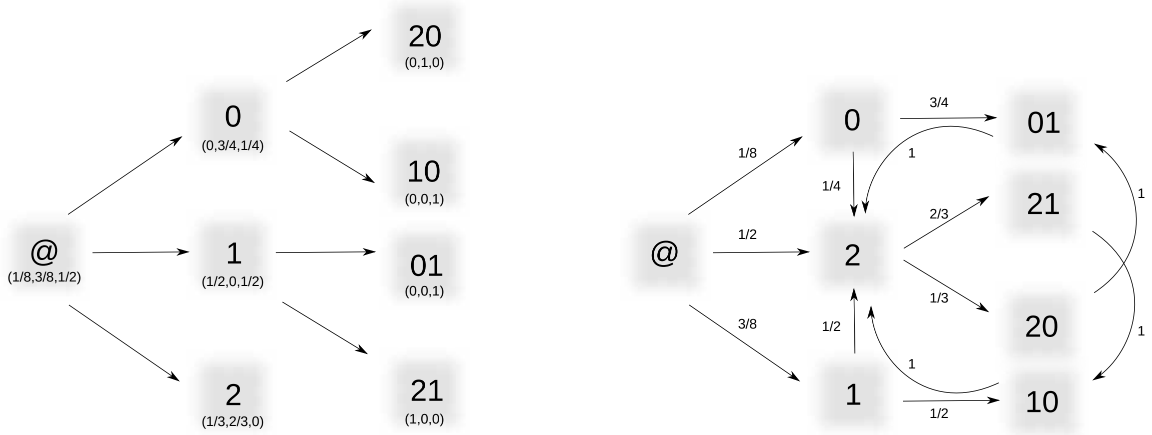

Figure 2: A transformation from a PST to a PSA [14].

2.3 T-Pattern recognition

The concept and algorithm behind T-Patterns was first stimulated by research regarding the structure of behavior and interactions with focus on real-time, probabilistic, and functional aspects, as well as hierarchical and syntactic structure, creativity, routines and planning.[15, 10] T-Patterns were chosen, because behavior patterns are often hidden in a behavioral stream and exist at different time scales.

The T-Pattern method described in [10] is an approach to find patterns in time series data, where causality between different data can be expected in short time intervals, relative to the observation period. This T-Pattern algorithm can be applied to sensor data in smart environments. The activity patterns in real sensor data are expected to be hidden in a data stream created by a large number of sensors. The approach can handle noisy, uncertain and incomplete sensor data, as the algorithm makes a probabilistic method available.[16]

The T-Pattern algorithm works with a bottom-up approach. A bottom-up approach means, that first the simplest T-Patterns are searched for and then, level by level, the higher hierarchical T-Patterns are found. These simple T-Patterns are at the fundamental level just simple pairs of sensor events having a statistical significant interval relationship.[15]

The assumption in the T-Pattern approach is a null hypothesis, expecting that each component is independently and randomly distributed over time with its observed average frequency. This null hypothesis is the assumption which all found T-Patterns are tested against.

The significance testing is based on Poisson probability distribution.

Making use of real-time information, the following probabilistic real-time relation serves as a key to the detection of hidden T-Patterns. If A is an earlier and B a later component of the same recurring T-Pattern, then, after an occurrence of A at t, there is an interval $[t+d_1, t+d_2], (d_2 \geq d_1 \geq 0)$ that tends to contain at least one occurrence of B more often than it would be expected by chance. This relation is called a *critical interval* (or CI) relation between the distributions of A and B. The later component B does not need to be the one immediately following A within the pattern, so even when some elements of a T-Pattern are ignored, it may still have the defining characteristics.

The behavior data are coded during an *observation period* $[1, N_T]$ in terms of discrete occurrence times of events. A T-Pattern can be seen as an ordered set $Q = X_1 \approx dt_1 X_2 \approx dt_2 \ldots X_m$, where X stand for simpler T-Patterns and each $\approx dt$ term stands for a relatively invariant (relative to the expectation assuming the null hypothesis) time distance between the components, which separates the consecutive X_i and X_{i+1} terms.[16] The algorithm as proposed in [10] computes if there is a series of intervals containing $k (k \geq 2)$ occurrences of A followed by at least one B and checks if k is significantly greater than the expectation, by varying the interval length for the search.

The input data are a set of sensor events, where timestamps represent the beginning and ending time of a specific sensor event. This data from real world settings are often sparse during long observational periods

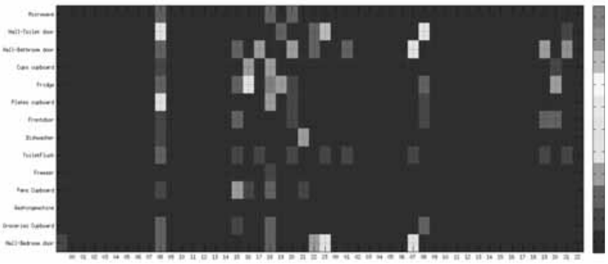

Figure 3: Sensor data recorded 2 days, accumulated hourly.

and clustered within short periods, which can be seen in Figure 3. This figure shows long periods without activity, and short periods with high activity. This problem can be solved by considered the time during the use of the T-Pattern approach. A small significance level leads to more reliable results as not less long patterns will be recognized ($\alpha = 10^{-8}$).

Some patterns appeared significantly more often than others, and those are essential for an activity to be recognized.

2.4 Suffix Trees and Probabilistic Finite Automata

Activity processes can be described by a subclass of probabilistic finite automata which are called Probabilistic Suffix Automata (PSA) by the authors of [4]. Probabilistic automata have similarities to widely used Hidden Markov Models, as analyzed in [17]. On the other hand PSAs seemed to be a logical extension to handle uncertainty in cases which can not be solved with pure finite state automata.

A PST T, over an alphabet Σ, is a tree of degree $|\Sigma|$. Each edge in the tree is labeled by a single symbol in Σ, such that from every internal node there is exactly one edge labeled by each symbol. The nodes of the tree are labeled by pairs (s, γ_s), where s is the string associated with the walk starting from that node and ending in the root of the tree, and $\gamma_s : \Sigma \rightarrow [0,1]$ is the next symbol probability function related with s. It is required that for every string s labeling a node in the tree, $\Sigma_{\sigma\in\Sigma}\gamma_s(\sigma) = 1$. A PST T generates strings of infinite length, but only probability distributions induced on finite length prefixes of these strings are considered. The probability that T generates a string $r = r_1 r_2 \dots r_N \in \Sigma^N$ is

$$P_T^N(r) = \prod_{i=1}^{N} \gamma_{s^{i-1}}(r_i),$$

where $s^0 = e$, the empty string, and for $1 \leq j \leq N-1$, s^j is the string labeling the deepest node reached by taking the walk corresponding to $r_i r_{i-1} \dots r_1$ starting at the root of T.

The PST is learned with a top-down algorithm by means of the sample statistic. With this T-Patterns the empirical probability function $\tilde{P}$ is defined and a suffix tree $\overline{T}$, which is a sub-tree of T with high probability, is constructed.

First, only a single node labeled as empty string e is constructed. Further nodes are added which are reasonably to be in the tree. This is the case when for the node v, the empirical probability of the suffix s, $\tilde{P}(s)$, is non negligible. Additionally, for σ the empiric probability of s followed by σ, $\tilde{P}(\sigma|s)$, must differ significantly from the probability of observing suffix(s) followed by σ, $\tilde{P}(\sigma|\mathit{suffix}(s))$. Suffix(s) is the suffix string of the parent node, meaning the way from the root to the parent node of v.

The decision rule is dependent on the ratio between $\tilde{P}(\sigma|s)$ and $\tilde{P}(\sigma|\mathit{suffix}(s))$. A given node is only added if this ratio is substantially greater than one. The tree is built level by level so the successors of a leaf are only added if they satisfy the condition. The tree reaches its final shape, if it does not fulfill the condition anymore.[4]

3 Results

In this section the results of the methods described so far are shown and discussed. The data from Kasteren *et al.* [13] are used to evaluate the T-Pattern algorithm and the results fulfills the demand. The last part focuses on the results from the algorithm constructing a PSA.

3.1 T-Pattern Analysis

The T-Patterns are evaluated with the annotated data in Matlab. In this analysis the significance test for the T-Patterns is done by the Binomial distribution. For each pattern the best fitting activity is searched for and is saved in the class with the highest percentage of overlap. These percentages are visualized in a pie and a histogram format.

In Figure 4, the histogram bars describes in 10% steps the percentage of patterns intersecting with an annotated activity. For better understanding 0 means no match with an activity, 1 means all patterns which

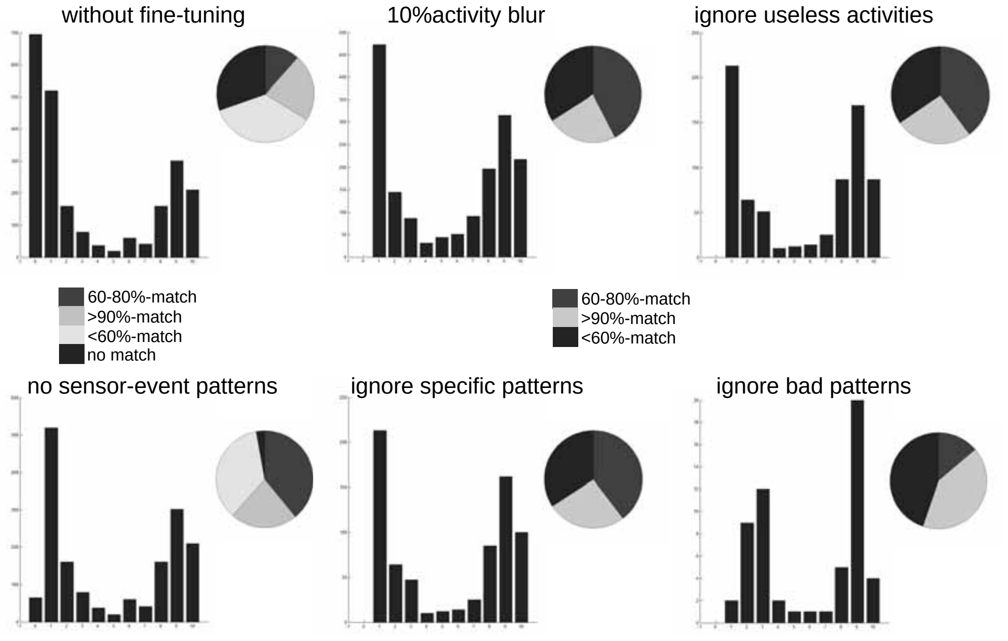

Figure 4: Fine Tuning.

intersect $0-10\%$ with an annotated activity, 2 means $10-20\%$ intersection and so on.

The pie has four different classes, 'no matches', 'more than 60%-matches', 'between 60% and 80%-matches' and 'more than 90%-matches'. This figures are all analyzed over the whole time period with the T-Pattern algorithm focusing on the sensor Ids 7 and 8. Sensor Id 7 stands for the 'toilet flush' and sensor Id 8 for the 'toilet door'. The figures must be seen as a continuous chain. All patterns which are left out in one picture are also left out in the other pictures in vertical direction.

In the first evaluation the T-Pattern algorithm finds too many T-Patterns compare picture one. This is the reason why a fine tuning of the recognized T-Patterns has been done to get more appropriate patterns. Especially the first two bars are a problem for further evaluation. This are those patterns which do not intersect with an activity or only intersect with at most 10%. In the first picture second row the T-Patterns which consist of only one sensor event are left out. A huge improvement can be seen especially in the 'no matches' part, as most of the individual single sensor-events are too small to overlap with an activity.

In the second picture the fine tuning is expanded by blurring the activities 10% each side. Other fine tuning steps like ignoring specific patterns, useless patterns and bad pattern lead finally to an appropriate result.

The significance analysis leads to the result of an improvement of while using lower significance level. The significance level 0.005 or 0.001 seems to be more appropriate than a bigger one like 0.05. The significance level is lowered with the consequence of less patterns being recognized. This patterns are those which match an annotated activity with higher percentage.

3.2 Probabilistic Suffix Tree and Probabilistic Suffix Automata

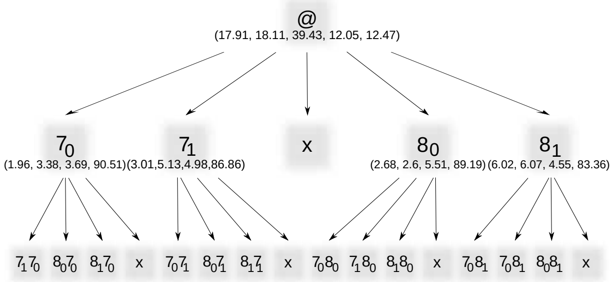

Figure 5: PST based on sensor 7 and 8 with pattern length 2.

In Figure 5 a result of the used system can be seen. In this case the sensors 7 and 8 are considered to build a PST. The probabilities are described in percentage,

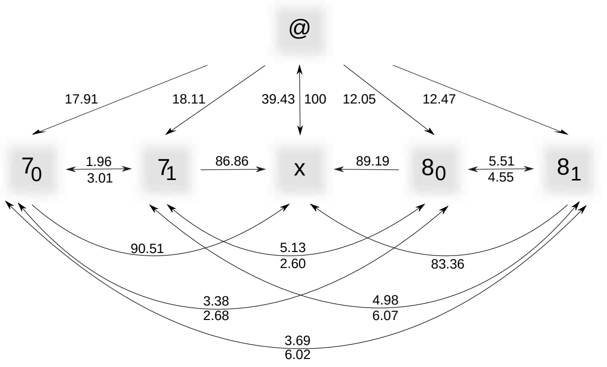

Figure 6: PSA transformed from PST in 5.

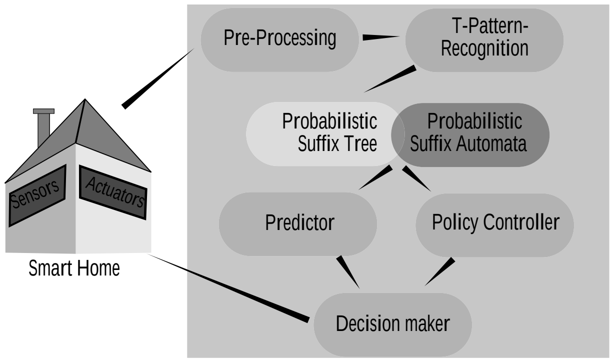

Figure 7: Activity recognition structure.

where the state 8_0 means 'toilet flush usage ended', 8_1 'toilet flush usage started', 7_0 'toilet door contact opened' and 7_1 means 'toilet door contact closed'. The results show that the probabilities are reasonable, as in each step the probabilities get smaller. This is of course true, because the occurrence of event A is at least as probable as the occurrence of pattern AB.

In Figure 6 the PST is transformed into a PSA with the already mentioned method. The result is an automaton, because for each state exists a subsequent state. If the activity, for instance 'using the toilet', is finished, the activity concludes with the state x, describing all events excluding the important ones 7 and 8. The two probabilities in Figure 6 assigned to arrows pointing in both directions describe the probability to the left node, indicated by the number above, and to the right node, indicated by the number below.

4 Conclusion and Outlook

The work describes a system which is able to deal with sequential data from domestic sensors. This sensor data are used to detect patterns and transform this patterns into knowledge, which can be further used to detect anomalies. This is done with the transformation of the T-Pattern sample statistic to a PST and further to a PSA. This PSA only differs in the arcs and not in the nodes or probabilities from the PST.

The algorithms described beforehand will fundamentally increase the efficiency of a home automation system in terms of configuration efforts due to automated processes. A conceptual structure of the overall system is depicted in Figure 7. The data from the Smart Home System from sensors and actuators are preprocessed, this means false data are deleted. After that the algorithm from above is applied to get a PSA.

The work ends with in this step, but further research can make it possible to detect anomalies on the basis of this algorithm. The idea behind is to use the prediction data from PSA and from policy controller, which contains user preferences and statistical data, for a decision maker. This decision maker, not yet enveloped, would be able to decide if a specific event occurred like leaving the Smart Home or optionally reminding the user to put on a jacket. The anomalies detection can also give hints about physical health status and diseases.[5]

References

[1] Nugent CD, Biswas J, Hoey J. *Activity Recognition in Pervasive Intelligent Environments (Google eBook)*. Springer. 2011.

[2] Han JW, Pei J, Yan XF. From sequential pattern mining to structured pattern mining: a pattern-growth approach. *Journal of Computer Science and Technology*. 2004; 19(3):257–279.

[3] Li K, Fu Y. Prediction of Human Activity by Discovering Temporal Sequence Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2014;36(8):1644–1657.

[4] Ron D, Singer Y, Tishby N. The Power of Amnesia : Learning Probabilistic Automata with Variable Memory Length. *Machine Learning*. 1996;25(2):117–149.

[5] Van Den Broek G, Cavallo F, Wehrmann C. *AALIANCE Ambient Assisted Living Roadmap*. Amsterdam, The Netherlands, The Netherlands: IOS Press. 2010.

[6] Mori T, Fujii A, Shimosaka M, Noguchi H, Sato T. Typical Behavior Patterns Extraction and Anomaly Detection Algorithm Based on Accumulated Home Sensor Data. In: *Future Generation Communication and Networking (FGCN 2007)*. IEEE. 2007; pp. 12–18.

[7] Albrecht DW, Zukerman I, Nicholson AE. Bayesian Models for Keyhole Plan Recognition in an Adventure Game. In: *User Modeling and User-Adapted Interaction*, 1. Springer Netherlands. 1998; pp. 5–47.

[8] Unsupervised Learning. In: *Advanced Lectures on Machine Learning*, edited by Bousquet O, von Luxburg U, Rätsch G, vol. 3176 of *Lecture Notes in Computer Science*. 2004;.

[9] Paz A. *Introduction to Probabilistic Automata*. New York: Academic Press. 1971.

[10] Magnusson MS. Discovering hidden time patterns in behavior: T-patterns and their detection. *Behavior research methods, instruments, & computers : a journal of the Psychonomic Society, Inc*. 2000;32(1):93–110.

[11] T-pattern analysis for the study of temporal structure of animal and human behavior: A comprehensive review. *Journal of Neuroscience Methods*. 2015;239(0):34 – 46.

[12] Fallmann S, Kropf J. Prediction of sequential patterns of domotic sensor data considering variable temporal pattern length. In: *WISHWell14 Proceedings- 6th International Workshop on Intelligent Environments Supporting Healthcare and Well-being*. Eindhoven. 2014; .

[13] van Kasteren TL, Englebienne G, Kröse BJA. Human Activity Recognition from Wireless Sensor Network Data: Benchmark and Software. In: *Atlantis Ambient and Pervasive Intelligence*, edited by Chen L, Nugent CD, Biswas J, Hoey J, vol. 4, pp. 165–186. Amsterdam: Atlantis Press. 2011;.

[14] Mazeroff G, De V, Jens C, Michael G, Thomason G. Probabilistic Trees and Automata for Application Behavior Modeling. In: *41st ACM Southeast Regional Conference Proceedings*. 2003; pp. 435–440.

[15] Magnusson MS. Repeated Patterns in Behavior and Other Biological Phenomena. In: *Evolution of Communication Systems: A Comparative Approach*, edited by Oller DK, Griebel U, pp. 111–128. Cambridge, MA: MIT Press. 2004;.

[16] Kropf J, Roedl L, Hochgatterer A. A modular and flexible system for activity recognition and smart home control based on nonobtrusive sensors. In: *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2012 6th International Conference on*. IEEE. 2012; pp. 245–251.

[17] Dupont P, Denis F, Esposito Y. Links between probabilistic automata and hidden Markov models: probability distributions, learning models and induction algorithms. *Pattern Recognition*. 2005; 38(9):1349–1371.