

Proceedings

A Domain Knowledge-Based Solution for Human Activity Recognition: The UJA Dataset Analysis [†]

Niklas Karvonen * and Denis Kleyko

Luleå University of Technology, Department of Computer Science, Electrical and Space Engineering, 971 87 Luleå, Sweden; denis.kleyko@ltu.se

* Correspondence: niklas.karvonen@ltu.se

† Presented at the 12th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI 2018), Punta Cana, Dominican Republic, 4–7 December 2018.

Published: 19 October 2018

Abstract: Detecting activities of daily living (ADL) allows for rich inference about user behavior, which can be of use in the care of for example, elderly people, chronic diseases, and psychological conditions. This paper proposes a domain knowledge-based solution for detecting 24 different ADLs in the UJA dataset. The solution is inspired by a Finite State Machine and performs activity recognition unobtrusively using only binary sensors. Each day in the dataset is segmented into: morning, day, evening in order to facilitate the inference from the sensors. The model performs the ADL recognition in two steps. The first step is to detect the sequence of activities in a given event stream of binary sensors, and the second step is to assign a starting and ending times for each of detected activities. Our proposed model achieved an accuracy of 81.3% using only a very small amount of operations, making it an interesting approach for resource-constrained devices that are common in smart environments. It should be noted, however, that the model can end up in faulty states which could cause a series of mis-classifications before the model is returned to the true state.

Keywords: data mining; activities of daily living; pattern matching; the UJA Dataset; binary sensors

1. Introduction

Recent advancements in sensor technology has made available small and inexpensive devices that can provide data about the user and its interactions with the surrounding environment. Analyzing this data can provide us with knowledge about user activity [1], which can be an important building block for developing context-aware pervasive and mobile applications. One such context is the activities of daily living (ADL) that can provide rich information about the user. Knowing what daily activity the user is performing allows us to analyze for example normal/abnormal behavior for health assessment, give timely feedback or advice to elicit behavioral change, or increase the efficiency of interaction with the application. This can have great potential in battling some growing societal problems, for example:

- Aging population—Applications that monitor the overall health of people can drastically reduce the impact on health-care services by allowing intervention only when needed (e.g., alarms) [2];
- Chronic diseases—By analyzing user behavior or biomedical signals, an application could detect the onset of chronic diseases [3] and possibly even prevent or slow progression of the disease if positive behavior change can be realized;
- Psychological conditions—Timely advice or interventions can be an important aspect in overcoming many psychological conditions such as fatigue syndrome, depression, and different forms of addictions [4].

This work was done during the participation in the 1st UCAmI Cup, which is an open competition in activity recognition. In this paper we analyze data from the UJAmI Smart Lab (More information

about the UJAmI Smart Lab can be found at: <http://ceatic.ujaen.es/ujami/en/repository>) in order to recognize a set of 24 different ADLs. The selected dataset represents 246 instances of 24 activity classes (see Table 1 in [5]) that were carried out by a single male inhabitant. The subject gave his informed consent for inclusion before participating in the study in accordance with the Declaration of Helsinki. The objective of the UCAmI Cup competition was to achieve as high accuracy as possible when classifying unseen sensor data. Competitors were given a labeled training dataset where the inhabitant performed activities over a period of 7 days. An unlabeled test dataset with sensor data for 3 days was also provided where competitors were supposed to label all 30 s time-slots using their solution. Each day in both datasets was divided into three segments: morning, afternoon, and evening. The dataset has four different data sources:

1. Event streams of binary sensors (see Table 2 in [5]).
2. Spatial data from an intelligent floor.
3. Proximity data between a smart watch worn by the inhabitant and Bluetooth beacons.
4. Acceleration data from the smart watch worn by the inhabitant.

In the solution described in this paper, only the event streams of binary sensors were used to recognize the activities. There were three reasons for this choice. Firstly, we consider binary sensors to be less intrusive than more sophisticated sensors. Secondly, they can typically provide low-noise data from which, if measured appropriately, strong inference can be made about activities (e.g., a toilet flush sensor typically is part of a toilet visit or a hygiene routine). Thirdly, the other sensor data was not found suitable for the chosen model (see Section 3 for more information).

The paper is structured as follows: in Section 2 we review related work in the field of ADL recognition, followed by information about the proposed solution in Section 3. Finally, we present our results in Section 4. They are followed by a discussion in Section 5 and conclusion in Section 6.

2. Related Work

A common approach to recognizing ADL is to craft different input features from sensor data and then apply machine learning. Tapia et al. [6] employed reed and piezoelectric switches in two different homes on various household elements and electronic appliances. They used a naive Bayes classifier to detect more than 20 activities. They found it possible to recognize activities of interest to medical professionals such as toileting, bathing, and grooming with detection accuracies ranging from 25% to 89% depending on the evaluation criteria used.

Singla et al. [7] investigate the use of probabilistic models to perform activity recognition of 15 ADL in a living lab setting with multiple residents. They used a Hidden Markov Model for AR and found that activities that were very quick (e.g., set table 21.21% accuracy) did not generate enough sensor events to be distinguished from other activities and thus yielded lower recognition results.

Krishnan et al. [8] used datasets from three living labs in order to evaluate different windowing techniques and window sizes when working with event-based sensors. They incorporated time decay and mutual information based weighting of sensor events within a window. Additional contextual information in the form of the previous activity and the activity of the previous window was also appended to the feature describing a sensor window. Their results suggested that combining mutual information based weighting of sensor events and adding past contextual information to the feature leads to best performance for streaming activity recognition.

3. Method

Activity recognition is a difficult task that involves careful choices of sensors and features, prediction method, filtering, relevant metrics for optimization and more. When data is difficult to interpret for a human, activity recognition is often performed using machine learning methods. In this work, however, we found that a domain knowledge-based solution was not only possible,

but it outperformed our initial experiments (not reported here) using some conventional machine learning methods.

3.1. Shortcomings of Conventional Machine Learning Methods for the UJA Dataset

A problem with applying conventional machine learning methods ([9], chapter 18) for classification, e.g., Naive Bayes, Decision Trees, Neural Networks, Support Vector Machines and others for performing recognition of ADL in the UJA dataset is their lack of a temporal state. Since static machine learning classifiers are essentially a map between an input vector of features and a set of output states, a certain input will always yield the same output. This makes it impossible for such models to learn temporal relationships between inputs over time, unless some “look back” is explicitly added as input features (e.g., a previous activity).

Recurrent Neural Networks (e.g., Reservoir Computing approach [10,11]), on the other hand, hold a temporal state and can learn temporal relationships (e.g., sequences) directly from input, making them a promising approach for activity recognition. Unfortunately, these methods are typically more complex which can make them require a significant amount of training data in order to perform well (e.g., a single-layered Long short-term memory network [12,13] with 100 units and an input size of 4 will hold 41,600 weight parameters in its model). This can make some common Recurrent Neural Networks perform poorly when datasets are small (the case of the UJA dataset) due to overfitting [14].

3.2. Motivation and Overview of the Manually Engineered Solution

Taking into account the nature of the UJA dataset and the shortcomings of the conventional machine learning methods, we decided to engineer the solution manually and tailor it to the ADL recognition. The following list summarizes the motivation for this decision:

- The dataset is event-based, therefore, it is non-trivial to modify the data into a continuous regular flow, which is assumed for the conventional machine learning methods with temporal states;
- The temporal nature of the dataset limits the applicability of the static feature-based machine learning methods;
- The limited size of the training dataset restricts the use of machine learning methods with temporal states. In an extreme case, an activity could be observed only once (Act11, Act12, and Act14) or twice (Act19 and Act21) in the training dataset;
- The data from the binary sensors can be interpreted by a domain expert in order to create rules for activity recognition.

Our manually engineered solution combined elements of both static and recurrent models in an attempt to remedy the previously mentioned shortcomings. It was chosen to approximate the behavior of the inhabitant using a rule-based approach that is very similar to a Finite State Machine (FSM). This is a reasonable assumption since such a model can be constructed even with small amount of data while it retains an internal state. Further, a FSM is easy for a human being to interpret in contrast to some machine learning methods, and in some applications (e.g., e-health solutions) the ability to understand the model can be important. It should be noted that an individual solution was developed for each segment, i.e., morning, afternoon, and evening segments were treated separately. However, the overall approach to the recognition is the same for every segment in the dataset and 3 different solutions can easily be merged into a single one (i.e., by using time as a first step to select one of the three models).

3.3. Choice of Sensor Data for Recognizing Activities

The provided UJA dataset included data from 30 binary sensors (see Table 2 in [5] for detailed description of the sensors). Upon inspection of the training dataset, it seemed that there were either some missing values or some missing labeling of the activities in the dataset. An example of missing

values is the door sensor (M01) in the afternoon segment of training day 6 which is expected during the enter the SmartLab activity (Act10) in the beginning of the segment. An example of a missing activity is the visit in the SmartLab activity (Act14) in the evening segment of training day 3. The ground truth for that segment does not include Act14, however, the door sensor (M01) was activated during the play a videogame activity (Act11). Since the activation of the door sensor during a leisure activity was found to be a good indicator of Act14, it was conjectured that Act14 was not labeled properly. Except for these errors, the binary sensors seemed to provide a solid basis for recognizing the activities.

When analyzing the training dataset we found that some sensors were only present in a certain activity, e.g., a washing machine is only triggered during “put washing into the washing machine” activity (Act20). We refer to such a sensor being a “signature sensor” for that activity. Further, there were sensors that were not unique to a certain activity, but that were always part of certain activities, e.g., the refrigerator sensor (D01) is often a part of prepare lunch (Act03) and prepare dinner (Act04) activities. We refer to those sensors as “descriptive sensors” since they are a part of the data representing an activity.

Although the binary sensors seemed robust and could provide strong inference on certain activities, some showed limitations in frequency over time making pattern recognition difficult (e.g., having a pattern of only one event). In order to complement the binary sensors, the use of RSSI values from the Bluetooth beacons to determine the closest Bluetooth beacon (by finding the minimum RSSI for a time slot) was evaluated. These data, however, revealed that distant objects at times appeared to be closest to the participant (probably due to the signal from closer objects being obstructed). During early experiments, this noise showed to impose a negative effect on the performance and this feature was, therefore, left out.

The provided dataset also had accelerometer data from the subject’s dominant wrist. While accelerometer data can hold rich information, using it for activity recognition typically requires extracting features from time or frequency domains using windowing techniques. Only a few activities (e.g., brush teeth (Act17), work at the table (Act21)) were assumed to have “signature movement patterns” that would be easy for a machine learning method to detect. Further, in order to use the accelerometer data in our state machine alphabet, it would have to be converted to a set of symbols. While this could certainly have been realized using a number of methods, this sensor data was chosen not to be included in the proposed solution.

3.4. Implementation

A transition table of the activities in the training dataset revealed that certain sequences of activities were always present. A representative example of such sequence can be found in the morning segments. During all 7 training days, the prepare breakfast activity (Act02) was followed by the breakfast activity (Act05), which in turn was followed by the brush teeth activity (Act17). Such patterns for each segment were extracted from the training dataset. This information was used as a temporal context and once the first activity of a sequence was detected the model was specifically looking for sensory information supporting the detection of other activities expected in that sequence.

The proposed solution performs the ADL recognition in two steps. The first step is to detect the sequence of activities in a given event stream of binary sensors. During the second step, the solution assigns starting and ending times for each of detected activities. Since each day in the dataset was split in three segments: morning, afternoon, and evening it was decided to construct an individual ADL recognizer for each segment. This choice was based on the fact that some activities were only present in a certain segment of the day (e.g., the breakfast activity (Act05) always occurred in morning segments). Also, it was not possible to differentiate some activities, e.g., the wake up activity (Act24) and the go to the bed activity (Act23) using only data from binary sensors, but many of such ambiguous activities were uniquely identifiable given the information about the segment type. Interestingly, given a segment type a descriptive sensor can become a signature sensor, e.g., the bed sensor (C14) is

descriptive for going to the bed and wake up activities. However, it becomes a signature sensor for wake up in A segments (morning) and going to the bed in C segments (evening).

3.4.1. Detection of an Activity Sequence

An FSM-based model was built (with IF-THEN rules in Matlab) using inference on the information from the binary sensors. The rules were designed and adjusted manually using the training dataset and were then tested on a small subset (fold). This method produced accurate results in terms of predicting correct sequences of activities.

3.4.2. Estimation of the Activities' Duration

Once an activity sequence was predicted, the next step was to estimate the starting and ending times of each activity in the sequence. This was approached in two different ways: for activities with signature sensors and for activities without them. It was easier for the activities with signature sensors since the times of occurrence of the first and the last signature sensory events were used as rough estimates of the starting and ending times. These estimates were then adjusted using offset constants. The offset constants were calculated from the training dataset as the average time between the events of signature sensors and the reported starting and ending times. A concrete example of this process is presented in Figure 1. Note that individual constants were acquired for each activity type. In the case of activities without signature sensors, their durations were estimated implicitly by using the predicted times for the neighboring activities and the estimates of the idle time intervals (gaps). These intervals were also calculated from the training dataset for each activity type. This case is graphically illustrated in Figure 2 using an example of the breakfast activity (Act05) in morning segments, which is always expected to be between the prepare breakfast activity (Act02) and the brush teeth activity (Act17).

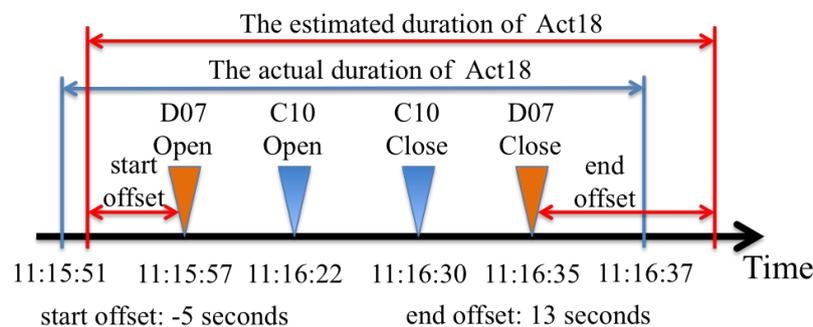


Figure 1. An example of estimating starting and ending times of Act18 in the morning segment of training day 1 in the training data. D07 and C10 are signature sensors for Act18. Note that the order of signature sensors in this case is not important, we only consider the first and the last signature events (orange).

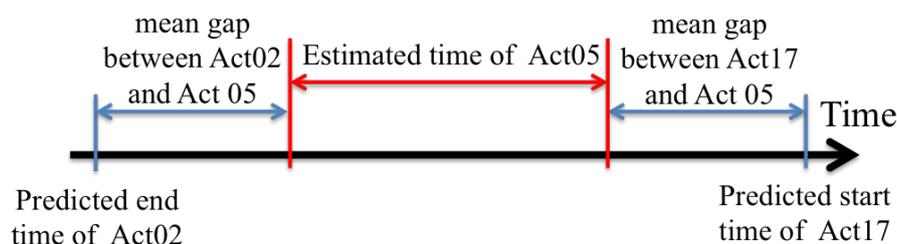


Figure 2. An illustration of an implicit estimation of starting and ending times of Act05 in the morning segments.

4. Performance Evaluation

The main performance metric used for assessing the quality of predictions was accuracy. The overall accuracy of the presented solution on the testing dataset was 81.3% while the average F_1 score and recall were 84.7% and 84.8% respectively. A more detailed performance of the model is provided in Table 1. It describes the accuracy of the solution for each day segment for all three testing days. As it can be seen from the table, the segment-wise accuracy varied between 38.5% (afternoon of testing day 3) and 93.2% (morning of testing day 3).

Table 1. Accuracy of the solution for each segment of the testing dataset

| Day Part | Day 1 | Day 2 | Day 3 |
|-----------|-------|-------|-------|
| Morning | 85.7% | 61.5% | 93.2% |
| Afternoon | 91.3% | 73.4% | 38.5% |
| Evening | 84.6% | 87.3% | 84.1% |

The main observation was that the second step of predicting starting and ending times of an activity was the main source of imperfections. In fact, the only error during the first recognition step (i.e., detection of activities in a data stream from binary sensors) was observed in the morning segment of testing day 2 where the work at the table activity (Act21) was missed. The occurrence of errors of such type can be attributed to the fact that for certain activities (e.g., Act21 or Act05) there are no signature sensors which would uniquely identify them in a data stream. Note that while these errors are rare they could significantly affect the performance since in the case of a missed activity, all time slots of that activity will be misclassified. For example, missing Act21 in the morning segment of testing day 2 resulted in 22 misclassified time stamps, which occupied 33.8% of the whole segment duration. Thus, this segment was the penultimate worst predicted segment in the whole testing dataset.

The other segments with the accuracy below the average (81.3%) were the afternoon segment of testing day 2 (73.4%) and the afternoon segment of testing day 3 (38.5%). The primary errors in these segments were caused by two different error types common to the second step of the solution. In the case of the afternoon segment of testing day 2, the main problem was the wrong prediction of the transition time between the prepare lunch activity (Act03) and the lunch activity (Act06), therefore, 15 time stamps were misclassified. The afternoon segment of testing day 3 is a good example of how errors in predicting the starting and ending times of an activity could affect the accuracy in a detailed time scale because the afternoon segment of day 3 was only 6.5 min (13 time stamps) long. It included three activities Act22 (6 time stamps), Act17 (2 time stamps), and Act13 (2 time stamps). Three time stamps were idle. The predicted sequence of activities was correct but the starting time of Act22 was delayed by 3 time stamps, which in turn contributed to the delay of Act17 prediction, which was 6 time stamps. Due to these delays there were only two out of six correct time stamps for Act22 and there were no correct predictions of Act17. Note also that these imperfections were the main reason of errors even in the segments with the highest accuracy (more than 90.0%) such as the morning segment of testing day 3 (93.2%) and the afternoon segment of testing day 1 (91.3%).

5. Discussion

The solution proposed in this paper deviates from the common approach of using conventional machine learning methods for activity recognition. It rather resembles an expert system designed to address problems in a particular domain. Nevertheless, and despite its simplicity, the solution achieved a fairly high accuracy of 81.3% in the classification problem with 24 different classes. Moreover, the misclassified time stamps occurred mostly either at the beginning or at the end of an activity due to imperfections in the predictions of the activity duration. For instance, this was the primary cause of the low accuracy in the afternoon segment of testing day 3. This drawback could potentially be

addressed by fusing sensory information for sources other than binary sensors, e.g., accelerometer data can be used for improving detection of starting and ending times of the brush teeth (Act17) activity.

Another potential shortcoming of the solution is that it could end up in a faulty state, i.e., not the state the user is in. This could cause periods of misclassification until the predictions and the actual activity are “in sync” again, which can happen either by chance or by a signature sensor firing. In other words, the current version of the proposed solution does not include a mechanism for recovering from a faulty state. Typical examples of a faulty state are transitions between an activity for preparing a meal (i.e., Act02, Act03, and Act04) and an activity for eating a meal (i.e., Act05, Act06, and Act07) since there are no solid indicators that one activity transitioned to another. This was the main reason for the deteriorated performance in the afternoon segment of testing day 2 where the transition between Act03 and Act06 occurred 15 time stamps earlier than the actual one and, thus, it was the cause of losing 19.0% of the accuracy for that day.

Finally, the idea of a FSM, which inspired the proposed solution, is an overly simplified model for the domain of interest since it assumes only one activity at a time. However, this simplified model is not unique to the proposed solution, but it is also common in activity recognition area as a whole. While this might not be a problem for certain applications, it limits the general applicability of the proposed solution.

6. Conclusions

In this paper we presented a domain knowledge-based solution designed to recognize activities of daily living among 24 different classes using only binary sensor data. During the design of the solution several assumptions were formed based on the training dataset. These assumptions included identification of typical patterns of activities in relation to a segment of a day as well as signature binary sensors which unequivocally identify certain activities. The training dataset was also used to fine-tune the parameters for assigning starting and ending time for each recognized activity. During the evaluation on the previously unseen testing dataset, the proposed solution achieved 81.3% accuracy. In our future work, we are going to augment the presented solution with the flow of events from data sources other than binary sensors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, L.; Hoey, J.; Nugent, C.D.; Cook, D.J.; Yu, Z. Sensor-Based Activity Recognition. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2012**, *42*, 790–808.
2. Wilson, D.H.; Atkeson, C. Simultaneous tracking and activity recognition (STAR) using many anonymous, binary sensors. In Proceedings of the International Conference on Pervasive Computing, Munich, Germany, 8–13 May 2005; Springer: Berlin, Germany, 2005; pp. 62–79.
3. Choi, E.; Schuetz, A.; Stewart, W.F.; Sun, J. Using recurrent neural network models for early detection of heart failure onset. *J. Am. Med. Inf. Assoc.* **2016**, *24*, 361–370.
4. Dumortier, A.; Beckjord, E.; Shiffman, S.; Sejdić, E. Classifying smoking urges via machine learning. *Comput. Methods Progr. Biomed.* **2016**, *137*, 203–213.
5. UCAmI Cup. Analysing the UJAEN Human Activity Recognition Dataset. 2018. Available online: <http://mamilab.esi.uclm.es/ucami2018/UCAmICup.html> (accessed on 17 October 2018).
6. Tapia, E.M.; Intille, S.S.; Larson, K. Activity recognition in the home using simple and ubiquitous sensors. In Proceedings of the International Conference on Pervasive Computing, Vienna, Austria, 14–17 March 2004; Springer: Berlin, Germany, 2004; pp. 158–175.
7. Singla, G.; Cook, D.J.; Schmitter-Edgecombe, M. Recognizing independent and joint activities among multiple residents in smart environments. *J. Ambient Intell. Humaniz. Comput.* **2010**, *1*, 57–63.
8. Krishnan, N.C.; Cook, D.J. Activity recognition on streaming sensor data. *Pervasive Mob. Comput.* **2014**, *10*, 138–154.
9. Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*; Pearson Education, Inc.: London, UK, 2010.

10. Lukosevicius, M.; Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **2009**, *3*, 127–149.
11. Appeltant, L.; Soriano, M.C.; Van der Sande, G.; Danckaert, J.; Massar, S.; Dambre, J.; Schrauwen, B.; Mirasso, C.R.; Fischer, I. Information processing using a single dynamical node as complex system. *Nat. Commun.* **2011**, *2*, 1–6.
12. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780.
13. Greff, K.; Srivastava, R.; Koutník, J.; Steunebrink, B.; Schmidhuber, J. LSTM: A Search Space Odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2222–2232.
14. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016. Available online: <http://www.deeplearningbook.org> (accessed on 17 October 2018).



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).