Experimental design for activity prediction in smart home environment

Gregor Donaj[1], **Mirjam Sepesy Maučec**[1], **Sašo Karakatič**[1], **Matej Borko**[2], **Andrej Žgank**[1]

[1] *Faculty of Electrical Engineering and Computer Science, University of Maribor*
[2]*A1 Slovenia*

E-mail: gregor.donaj@um.si, mirjam.sepesy@um.si, saso.karakatic@um.si, matej.borko@a1.si, andrej.zgank@um.si

Abstract

In this paper, we design an activity prediction framework to be used in a smart home environment. Because the data from target environments is not yet available, we propose an approach of using an open dataset and adopting it for the target environment. Two machine learning techniques were studied, Random Forest and HMM. Based on preliminary experiments, HMM provide better results and we decided to study it further in the future.

1 Introduction

Progress of technologies in machine learning and pervasive computing has generated interest in the development of smart environments that will assist humans with many valuable functions to improve living. A smart home is the most widely studied smart environment [1]. It has been researched for nearly a couple of decades, and it is even more intensive researched today. The main reason for increasing amounts of attention is the rising aging population. The concept of smart homes gives elder people hope to extend the period living independently within their home environment.

The idea of a smart home, stated more broadly, is to support the well-being of the residents of the home. Our goal, presented in this paper, is a prediction of residents' activity in the smart home environment. To develop the prediction system large and varied data sets need to be collected over time, as they are necessary to uncover hidden patterns and unknown correlations between data. Collecting them is expensive and time-consuming. In this paper, we propose an approach of using an open dataset and adopting it for the target environment.

The paper is organized as follows: Section 2 discusses related work. In Section 3 we describe the general features of CASAS open dataset that was chosen for transformation to our target environment. In Section 4 we describe the target environment and specify how the data from CASAS dataset was mapped to the foreseen data in the target environment as well as enriched with third-party environmental data. In Section 5 an overview of methods, commonly used for activity prediction is given. For our practical work, we selected two of them and performed initial experiments. We report the preliminary results. Section 6 concludes the paper. We give our plan for future work on the project.

2 Related work

Currently, numerous Smart Home projects intend to make daily life comfortable. Some of them were developed for proof-of-concept demonstration, and others were integrated into real life environments.

In [2] learning techniques were developed to discover patterns in the resident's daily activities and to generate automation policies that mimic these patterns. Ontological modeling and semantic reasoning were integrated into an agent-based system for activity recognition [3]. Data-driven approaches for predicting future activities shown that regression learners could be useful to learn an activity predictor that can reason about relational and temporal structure among the activities and to forecast future activity occurrences [4].

A prerequisite for the research on activity prediction is the availability of datasets which are large enough and ready to be used and analyzed. Because no data has been collected in our target environment yet, we have searched for open datasets available for public use. GCDC [5] is the collection of data about users cooking meals. It does not contain information about other activities. In d-WAR [6] datasets, users were wearing sensors, and the data was sent by the mobile phone to the base station. There were no stationary sensors. DRED [7] includes only sensor data, whereas activities are missing. The MIT [8], CASAS [9] and ARAS [10] datasets contain data about stationary sensors and activities. We decided to use CASAS data collection because the experimental environment in this project is most similar to our target environment.

3 CASAS dataset

The CASAS project took place at Washington State University [11]. The idea was to design a "smart home in a box" that is ready to be used to perform key functions out of the box. The CASAS data collection contains 32 smart home datasets: 19 datasets represent single-resident sites, 4 represent locations with two residents, and the rest house larger families or residents with pets. Most activity recognition algorithms have been tested in situations where a single individual is performing activities in a continuous, sequential manner. CASAS project focuses on a different complexity issue for home behavior, that of recognizing

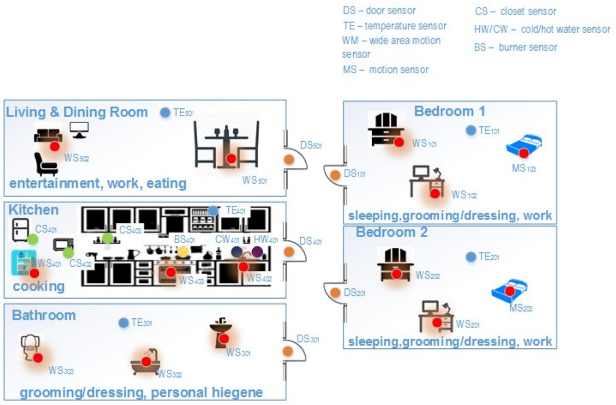

Figure 1: The target environment.

activities of more than one resident. The data collection includes 20 residents aged 21 to 62 years (mean 33 years), 8 males and 12 females, with a variety of background and technological familiarity [11].

In the CASAS project, the status of the residents and their physical surroundings are perceived using sensors. The examples of sensors are motion sensors, door sensors, temperature sensors, etc. Sensors generated events that consisted of a date, a time, a sensor identifier, and a sensor message. Sensor data were captured using a customized sensor network and stored in a database. Data was mapped to corresponding activity labels by using an activity recognition software, capable to label activities in real-time as sensor events arrive in a stream. The examples of activities are: bed-toilet transition, cook, eat, enter home, leave home, personal hygiene, etc. To maintain privacy the participant names and identifying information were removed and the collected data was encrypted before it was transmitted over the network.

In our experiments, only the following CASAS datasets were used: 7, 8, 11, 14, 15, 17 and 35. In datasets 7, 8, 11 and 14 one resident was living in an apartment, and in the rest, two of them.

4 Mapping from CASAS to target environment

The first step in building the dataset was to collect the appropriate files in the selected CASAS datasets, transform them into a consistent format, and make a preliminary analysis of the datasets. This analysis was necessary to define an appropriate target environment onto which the CASAS datasets could be mapped.

We also corrected some obvious errors (e.g., correcting the year from 22009 to 2009) and removed inconsistencies between the datasets (e.g., different column delimiter characters in datasets) to be later able to build a concise single-file dataset.

The mapping procedure is intended to construct a modified and combined dataset, which we call the IQdatabase, as this work is part of the IQ Home Program (http://www.iq-home.si/en/).

Table 1: Excerpt from the sensor mapping rules.

ID	Original sensors	Target sensors
7,8,11	D005, M37	DS301
7,8,11	M38	WS301
7,8,11	M39	WS302
7,8,11	M40,M41	WS303

Table 2: Excerpt from the activity mapping rules.

Original activity	Target activity
R1_Meal_Preparation	meal_preparation
R1_prepare_dinner	meal_preparation
R1_prepare_lunch	meal_preparation
R2_Meal_Preparation	meal_preparation

4.1 The target environment

After examining the selected datasets, we designed a custom target environment into which all data from the used CASAS datasets were mapped. The target environment is presented in Figure 1. The number and type of rooms are determined to fit the CASAS datasets. The most significant difference from the CASAS datasets is the use of a hallway with doors to all other rooms, while in the CASAS apartments some rooms were directly connected.

The sensors in the target environment have two-letter designations based on the type of sensor, eg. DS for door sensor (see figure), followed by a three digit number, where the first digit identifies the room. This way we were able to define a more consistent designation system compared to the original CASAS datasets.

4.2 Mapping procedures

The next step was to determine the mapping rules for sensors and activities. Examining the environments from the original CASAS datasets, the sensor data associated with different activities, the position of sensors, and the most likely rooms for activities, a mapping scheme was determined. A part of it is shown in Table 1 for sensor mappings and Table 2 for activity mappings.

In Table 1 we see that some rules are defined for several CASAS datasets, as they were collected in the same apartment. In Table 2 we see that some activities were combined into one, as not all original datasets had the same activities. Also, activities for both residents (R1 and R2) were mapped into one activity. Some activities were not mapped at all as they appear only in one of the datasets.

The original datasets, as well as our target environment dataset, are all text files. We, therefore, used simple Perl wrappers to perform all preprocessing and mapping.

4.3 Enriching data with environmental information

The context of the activities are the external data (besides the actions) that can be gathered from various sensors around or inside the home. CASAS datasets do not have any external information in their provided form, so we were forced to use third-party sources for this task. The literature of research on this topic shows that there is a connection between subject actions and the weather and environment data inside and outside the monitored home.

For this reason, we wanted to enrich the CASAS data with external environment data from third party sources, to provide more valuable information, that can later be used in the pattern recognition process with artificial intelligence.

After extensive research, we decided to use weather information from Dark Sky service [12], where they offer free API to gather historical weather information for almost any location in the world. Their limitations on the API calls were inside of our call numbers, so there was no problem with this, and their terms of using their data also suited our type of work.

For gathering data from Dark Sky API, we used Python 3, and we used Python wrapper by Ze'ev Gilovitz [13], which provides an easy way to communicate with Dark Sky API with Python programming language.

After surveying the literature on the topic, we concluded to enrich CASAS data with the following environment data: Temperature (outside temperature in Celsius unit), summary (written summary of the weather conditions), icon (icon code used for the weather maps – can be used as an indicator of weather conditions), humidity (percentage of the outside humidity), air pressure (in millibars) and wind speed (in miles per hour).

4.4 The IQdatabase

After mapping, all data were combined into a final dataset, which we called IQdatabase (currently version 2.0). The database is formatted as a single text file with 48 columns per line. The columns indicate the original CASAS dataset number, date, time, the current status of 32 sensors, the state of 6 environmental data, and the current status of 7 activities, namely: work, dressing, personal hygiene, eating, meal preparation, sleeping, and TV.

To be able to perform experiments, the database was divided into a train set, a development set, and a test set. Both the development and train sets contain about a month of data and are taken at the end of the original dataset number 11, which is the largest original dataset.

The train set was further divided into 3 parts. The first part includes all other data from dataset 11. The second includes all data from other datasets from the same apartment as dataset 11. The third dataset includes all data from other locations. The sizes in terms of lines of data for all sets are given in Table 3.

Table 3: The data sizes in the IQdatabase.

Set	Size
Test	211,722
Development	194,220
Train 1	1,210,515
Train 2	663,363
Train 3	2,255,638
Train - total	4,129,516
Total	4,535,458

5 Activity prediction

Models, used for activity prediction, can be classified as hand-crafted, logic based or probabilistic. In hand-crafted models, training data is manually examined to find patterns that are used to predict activities. Logic-based models automatically learn rules from training data. The learning system is based on the principles of inductive logic programming (especially search, representations, operators, background knowledge) with transformation-based tagging (e.g., error-driven search).

Hand-crafted and logic-based models are deterministic. As activities are commonly performed in a non-deterministic way, probabilistic models are the most popular approaches for activity prediction. These models take into account that sensor readings in a real smart home are sometimes noisy. Several probabilistic models have been proposed to model the sequence classification problem of activity prediction: naïve Bayes classifiers, hidden Markov models, support vector machine, artificial neural networks, conditional random fields, random forest classifier, etc. In our research, we decided to use Random Forest (RF) and hidden Markov models (HHM). Both of them were successfully used in other researches [14].

5.1 Predicting with Random Forest

The Random Forest was proposed in [15] and further developed in [16] and is an ensemble learning algorithm, used for classification and regression problems. An RF classifier consists of a group or ensemble of decision tree predictors, each capable of producing a result when presented with a set of previously unknown data. The result of the decision tree takes the form of one of the predefined classes. The ensemble of decision tree results are aggregated, and the most popular class is chosen.

The main difference between the bag of decision trees and the RF method is that the bag of decision trees combines the result of independently constructed decision trees each on their own randomly selected subset of training data, but RF method also changes the way each decision tree in the ensemble is constructed. In the construction of one decision tree for the RF ensemble, each process of making a decision tree node consist of an additional step - the independent predictor value choosing for that particular node is done on a random subset of predictor values. Besides the random sampling of training data for each decision tree present in the creation of the decision tree, this step of randomly subsetting of available predictors ensures that more diverse trees are constructed and widen the search space for the classification prediction.

The problem of activity prediction is a time series problem, where prediction is made on the current information and the past data. Thus, we had to preprocess the available data in such way that every instance should also include information from the past, and so regular classi-

Table 4: Results of activity prediction in the IQdatabase.

	Precision	**Recall**	**F1-score**
Random Forest			
Train 1	0.75	0.26	0.27
Train 1+2	0.35	0.28	0.19
Train 1+2+3	0.25	0.20	0.14
HMM			
Train 1	0.47	0.21	0.15
Train 1+2	0.42	0.16	0.11
Train 1+2+3	0.45	0.18	0.14

fiers could perform prediction. This was done with the sliding windows procedure.

For our experiment, we used the following settings: window size: 10 past events; RF size: an ensemble of 100 decision trees in the forest. We used the default classifier output.

5.2 Predicting with Hidden Markov Models

Hidden Markov Models (HMM) are an extension of a Markov chain random process, hidden from the observer, and an observable random variable depended only on the current state of the Markov chain. HMMs are used to model processes, where the system has only input data, available through a noise channel. The most known example of using HMMs is acoustic modeling in speech recognition [17].

In our system, the hidden states represent the current activity in the environment. One possible state is also the absence of any activity. The Markov chain is the model for transitions between activities. The observable random variable is the state of all sensors and environmental data.

Since our training sets contain all activity and sensor data, it is simple to train the HMM. The Markov chain is directly trained based on all transitions between activities. The observable variable is trained as a set of independent variables. Most of them have discrete values, e.g., a sensor is ON in OFF.

5.3 Results

We performed two series of experiments. The first was based on Random Forest and the second one on HMM. In both series, the experiments were repeated three times, each time with a different training set. The results of preliminary experiments are collected in Table 4. Precision was selected to be the key metric. When Using all training material, HMM brought more promising results. We will continue the research with HMM in the future.

6 Conclusion

In this paper, we have proposed an approach to adopting an open dataset for activity prediction in smart home environments, for which the data is not yet available. HMMs have shown promising results for activity prediction and will be studied in more details in the future. Our next step is also to conduct Wizard of Oz experiments in real-life environments.

The application of these methods will be further applied in smart home environments for the IQ Home project, which is meant for establishing a supportive environment for smart home residents.

Acknowledgment

This work was partially funded by the Republic of Slovenia and the European Union under the European Regional Development Fund – IQ Home Program “Intelligent home of the new generation designed on smart appliances and wood”.

References

[1] L. C. De Silva, C. Morikawa and I. M. Petra. State of the art of smart homes. Engineering Applications of Artificial Intelligence, 25(7), 1313-1321, 2012

[2] P. Rashidi and D. J. Cook. Keeping the resident in the loop: Adapting the smart home to the user. IEEE Transactions on systems, man, and cybernetics-part A: systems and humans, 39(5), 949-959, 2009.

[3] L. Chen, C. D. Nugent, and H. Wang. A knowledge-driven approach to activity recognition in smart homes. IEEE Transactions on Knowledge and Data Engineering, 24(6), 961-974, 2012.

[4] B. D. Minor, J. R. Doppa, D. J. Cook. ”Learning Activity Predictors from Sensor Data: Algorithms, Evaluation, and Applications”, IEEE Transactions on Knowledge and Data Engineering, vol 29, no. 12, 2744-2757, 2017.

[5] GCDC, http://kitchen.cs.cmu.edu/ [Dostopano: 21.8.2018]

[6] d-WAR, https://people.eecs.berkeley.edu/˜yang/software/WAR/index.html [Dostopano: 21.8.2018]

[7] DRED, http://www.st.ewi.tudelft.nl/˜akshay/dred/ [Dostopano: 15.2.2018]

[8] MIT DB, http://courses.media.mit.edu/2004fall/mas622j/04.projects/home/ [Dostopano: 21.8.2018]

[9] CASAS, http://ailab.wsu.edu/casas/datasets/ [Dostopano: 21.8.2018]

[10] ARAS, https://cmpe.boun.edu.tr/aras/download.php [Dostopano: 21.8.2018]

[11] D. J. Cook, A. Crandall, B. Thomas, and N. Krishnan. CASAS: A smart home in a box. IEEE Computer, 46(6):26-33, 2013.

[12] DarkSky, https://darksky.net/dev/ [Dostopano: 21.8.2018]

[13] Python API, http://zeevgilovitz.com/python-forecast.io/ [Dostopano: 21.8.2018]

[14] E. Nazerfard, B. Das, L. B. Holder, and D. J. Cook. Conditional random fields for activity recognition in smart environments. In Proceedings of the 1st ACM International Health Informatics Symposium, pp. 282-286. ACM, 2010.

[15] T.K. Ho. The random subspace method for constructing decision forests. IEEE transactions on pattern analysis and machine intelligence, 20(8), 832-44, 1998

[16] L. Breiman. Random forests. Machine learning, 45(1), 5-32, 2001

[17] M. Gales, S. Young. The Application of Hidden Markov Models in Speech Recognition. Foundations and Trends in Signal Processing, 1(3), 195-304, 2007