Per Lynggaard

Distributed smart home activity recommender system using hidden Markov model principles

Ekonomiczne Problemy Usług nr 104, 359-369

2013

ZESZYTY NAUKOWE UNIWERSYTETU SZCZECIŃSKIEGO

NR 762 EKONOMICZNE PROBLEMY USŁUG NR 104 2013

PER LYNGGAARD
Aalborg University

DISTRIBUTED SMART HOME ACTIVITY RECOMMENDER SYSTEM USING HIDDEN MARKOV MODEL PRINCIPLES

Introduction

The availability of smart homes will have a huge impact on our future lifestyle because they will be able to act "intelligently" and provide services according to our personal preferences. So, smart homes may take care of and communicate about a lot of tasks, for example regulating power consumption as a function of time and controlling light, ventilation and heating systems to fulfil the user's needs[1].

Many home automation systems often used the strategy of centralizing the smart object event and action processing[2]. This approach has some disadvantages, such as severe security flaws, single point of failure sensitivity and the requirement for a large amount of network bandwidth and processing power[3]. Another design strategy is distributing the smart home event and action processing between different smart home systems running in parallel. This strategy reduces the impact of the security flaws and single points of failure. It also offers the possibility of pre-processing the smart object events and actions before transmitting them to the next part of the distributed processing chain. This is an enabler for a variety of different

---

[1] K. Balasubramanian, A. Cellatoglu: *Improvements in home automation strategies for designing apparatus for efficient smart home*, Consumer Electronics, IEEE Transactions on, Vol. 54, No. 4, 2008, pp. 1681–1687.

[2] Thinagaran Perumal, A R Rmali, Chui Yew Leong: *Interoperability Framework for Smart Home Systems*, IEEE Transactions on Consumer Electronics, Vol. 57, No. 4, 2011, pp. 1607–1611.

[3] M Starsinic: *System Architecture Challenges in the Home M2M Networks*, in Applications and Technology Conference (LISAT), Long Island Systems, 2010, pp. 1–7.

smart object types that offer different processing power and network resources[4]. Some of these could e.g. be combined into a distributed smart home system swarm using the concept of embedded cloud computing[5].

This paper presents a distributed smart home system that is divided into two blocks named: Smart Home System part I (SHS-I) and Smart Home System part II (SHS-II). SHS-I is a distributed artificial intelligence system that are embedded into the smart home devices. It offers a simple system that are able to learn and predict stateless user actions as e.g. making breakfast, taking a shower, etc. The SHS-II is basically a system that is able to learn an activity from a sequence of user initiated actions and based on this, proposes a correlated smart home activity to the user.

This work is organized as a presentation of the smart home model followed by a discussion of the involved blocks and components. The SHS-II component is investigated and discussed at a very detailed level. First, the theoretical framework is presented. Second, its performance is validated by implementing a Java based simulation model that is able to learn from a sequence of activities and predefined user-annotated activities.

1. Related work

Many related papers have been issued about artificial intelligence in smart homes, some of these use probabilistic models as the basis for detection. Especially, the work done by Kasteren et al.[6] is relevant. They worked with a simple sensor network approach in combination with both an HMM and a conditional random fields (CRF) model for classification. Their work has achieved good accuracy for activity prediction, that is, training and prediction on the fly, but it suffers from a high degree of complexity, that is, high performance loss in running the full algorithms, and it does not provide the flexibility and advantages a distributed smart home system offers. Fang et al.[7] have tested the Näive Bayes, HMM and Viterbi algorithms with respect to detecting human activities from observed sensor events. They have used the huge CASAS data set and looked into the effect of different time window lengths. Their findings are that a quantized time window, quite similar

---

[4] S. Bhardwaj, T. Ozcelebi, J. Lukkien, and C. Uysal: *Resource and Service Management Architecture of a Low Capacity Network for Smart Spaces,* IEEE Transactions on Consumer Electronics, Vol. 58, No. 2, 2012, pp. 389–396.

[5] X Ye and J Huang: *A Framework for Cloud-based Smart Home*, International Conference on Computer Science and Network Technology, 2011.

[6] T.V. Kasteren, A. Noulas, G. Englebienne, and B. Kröse: *Accurate Activity Recognition in a Home Setting,* UbiComp ’08, September 21–24, Seoul, Korea, 2008.

[7] H. Fang, R. Srinivasan, and D.J. Cook: *Feature Selections for Human Activity Recognition in Smart Home Environments*, International Journal of Innovative Computing, Information and Control, Vol. 8, No. 5, May 2012.

to the one used in this work, should have a length of one hour and it should cover a time span of either 12 or 24 hours to achieve optimal learning. So, from this result, the number of quantization steps in this work has been selected as 24, each of one hour duration. A paper by Cook[8] discusses if it is possible to generalize activity learning over different environmental settings and resident types. She concluded that this is possible to some extent. So, her work supports the assumption that an action sequence contains correlation information, as also assumed in this work. Furthermore, this conclusion is also very promising in the light of offering "out of a box" smart home technologies.

2. Model and system architecture

A model that covers the presented smart home system at a high abstraction level is illustrated in Fig.

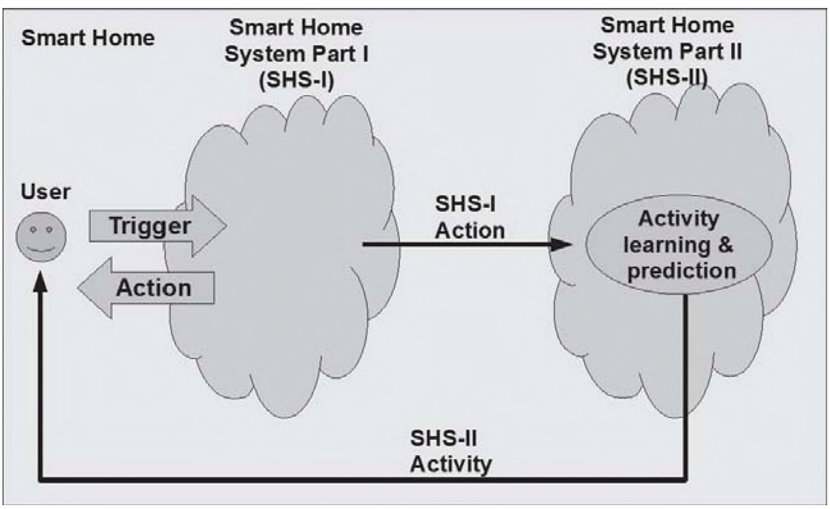

Fig. 1. A model for the presented smart home system

Source: own elaboration.

Leftmost in Fig is a smart home user that carries out scenarios in the form of a normal living pattern in a smart home. This way the user interacts indirectly with the smart home by triggering sensors and receives feedback in form of actions. These actions are carried out by the smart home system actuators. In the presented

[8] D.J. Cook: *Learning Setting-Generalized Activity Models for Smart Spaces*, Intellligent Systems IEEE 2012, Vol. 27, No. 1, pp. 32–38.

smart home system these sensors and actuators are assigned the conceptual name Smart Home System one (SHS-I). Such a system is limited by the low amount of processing power available in the small embedded microcontrollers and by the available battery power source, etc. To overcome these limitations an advanced and extended smart home system is added in form of a more sophisticated artificial framework. This framework is named Smart Home System two (SHS-II) to indicate that it is an add-on to SHS-I. The SHS-II cannot be used as a standalone system because it require the predicted actions from the SHS-I system or from some other compliant systems (e.g. Naïve Base based), as input. Thus, is uses the actions from SHS-I to make its own prediction.

This SHS-II system is the target for this paper, so the next section will describe this from a more technical point of view.

3. Activity processing (SHS-II)

The activity processing in the SHS-II system is illustrated in Fig.

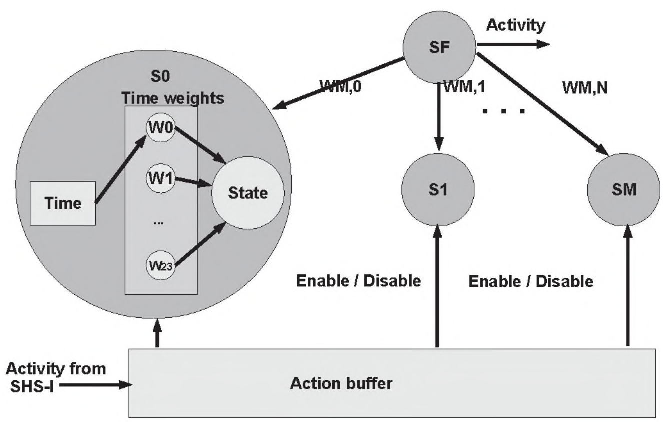

Fig. 2. Activity processing in SHS-II

Source: own elaboration.

Actions arriving from SHS-I are placed in an action buffer in the time order of arrival. Old actions in the action buffer that are beyond a predefine time limit are simply thrown away in a cyclical manner so the newest actions are always placed first in the buffer.

These actions are processed by traversing the action buffer. First, the action name is used as a key to enable the respective state using the Enable / Disable flow in Fig. When a state is enabled, the action arrival time is used as an index that points out one specific weight in a time pool of weights, see Fig. Thus, each state contains weights that express an un-scaled probability (the term un-scaled "probability" means it does not sum to one) for the action to happen at that time. These weights quantize time into intervals of one hour. In total this means there are 24 of them, because there is no date or year information type provided. So, this system offers the preservation of action arrival sequence and arrival time in the states.

Looking at Fig the similarity to the HMM is visible, especially if the action buffer is modelled as the observable variable and the states as a hidden variable. The fact that the HMM offers a relaxation of the independent and identically distributed (i.i.d.) assumption often used to simplify classifiers, means that cross-correlation between the actions can be handled. Often, a huge matrix is required to handle this, but using the Markov assumption it can be assumed that future predictions are independent of all but the most recent observations. Based on this assumption, the presented SHS-II agent captures a specific predefined activity, for example "set home into sleep mode", in state S_F that is dedicated to look for this action only. Then the system uses the time buffer in combination with the previous state candidate and the time based weights to look up the most likely last state (i.e. a hidden state estimate).

The targets in the following sections are the learning and prediction principles that are extracted from the more general HMM and Viterbi theory.

4. Activity learning and prediction

The HMM based learning process used in this work is iteratively based, that is, the model learns "on the fly" in the form of online learning. This is achieved by updating the estimated joint probability whenever the user supplies a predefined activity that matches.

From a statistical point of view it is assumed that the y-vector contains the predicted activity and the x-vector contains the input from the actions in the action buffers in the time window T. Thus, the joint probability for the suggested system can be expressed as: $p(\overline{x}_n, y_n) = p(\overline{x}_n|y_n)p(y_n|y_{n-1})$ where n is the discrete quantized time. Looking into the right hand side, the first factor and using the defined x- and y-vector nomenclature, the conditional probability for x being the vector that produces activity in time window T can be described by a simplified Bernoulli distribution. The parameters for this distribution can be estimated analytically using a maximum likelihood parameter estimation method. Regarding the right hand side second factor, it expresses the transition probability distribution that represents the probability of going from one state to the next. These transition probabilities can be represented by well-defined a multinomial distribution.

Regarding the activity prediction it is performed by using a simplified probabilistic Markov classification approach combined with a simple threshold. As already stated, it is assumed that all actions are temporally related. So, regarding the inference problem, it deals with finding the best single state sequence that maximizes the probability $p(\bar{x}_n, y_n)$. One often used efficient strategy is the Viterbi algorithm, because it reduces the calculation's complexity. Combining this with the first order Markov assumption reduces the buffer look back. A simplified Viterbi optimization process where it is assumed that the initial probability is equal for all states and the ending state is the predefined agent state S_F is given by $\lambda_M = [\max_{1 \le i \le M} p(\bar{x}_{n-1} | y_{n-1} = i) a_{ij}] p(\bar{x}_n | y_n = S_F)$ which states that given y_n is the specific predefined action (i.e. action i) and the vectors x_{n-1} and x_n are the given observations, the highest probability must be searched by varying the choice of the previous state and its transition probability to state i.

From this expression it is observed that the maximal probability is found by performing a search through all the previous states multiplied by their transition probability to the current predefined state S_F. Thus, the prediction phase takes place as follows. When a new action arrives in the action buffer, the buffer is traversed by processing the actions one at a time. So, based on each action, the related state S_m and the time quantized weight are located. This weight is then multiplied with the transition weight connecting that state to the specific predefined action S_F. At the end of the cyclical process, the highest value is found and multiplied with the weight.

After using a threshold limit, the value that exceeds this limit is selected as the best estimate for the predicted current activity. Note that the threshold process restricts the importance of the selected activity and, thereby, whether it is presented for the user.

5. Implementation and performance

The described smart home system is modelled using a Java program running on a common PC. All the essential algorithms are implemented on this platform. Parameter settings for these algorithms have been chosen by using an experimental approach.

To test the SHS-II system, the Aruba 2010–2011 data set from the WSU CASAS smart home project[9] has been used. This was recorded in a house with 26 sensors where a woman lived for approximately 7 months. The woman's children and grandchildren visited on a regular basis. This resulted in 6468 sensor events that are all annotated by the user. These events are: meal_preparation, relax, eating,

---

[9] D.J. Cook: *Learning Setting-Generalized Activity Models for Smart Spaces*, IEEE Intelligent Systems 2012, Vol. 27, No. 1, pp. 32–38.

work, sleeping, wash_dish, bed_to_toilet, enter_home, leave_home, housekeeping and resparate.

To be able to test the SHS-II system it has been defined that a "leave home" and "enter home" difference of more than 1 hour should be detected. This enables the SHS-II system to autonomous power down the home when the user has left, etc.

As discussed earlier, it is a requirement that correlation between the users activities exists in order to achieve good system performance. Thus an experiment is performed to clarify if this is the case in the used data set. By running the CASAS data through the learning algorithm triggered by the "leave home" activity and mapping the weights, results. It should be noted that these data have been pre-processed so only those where the user is away for more than 1 hour have been used in this figure.

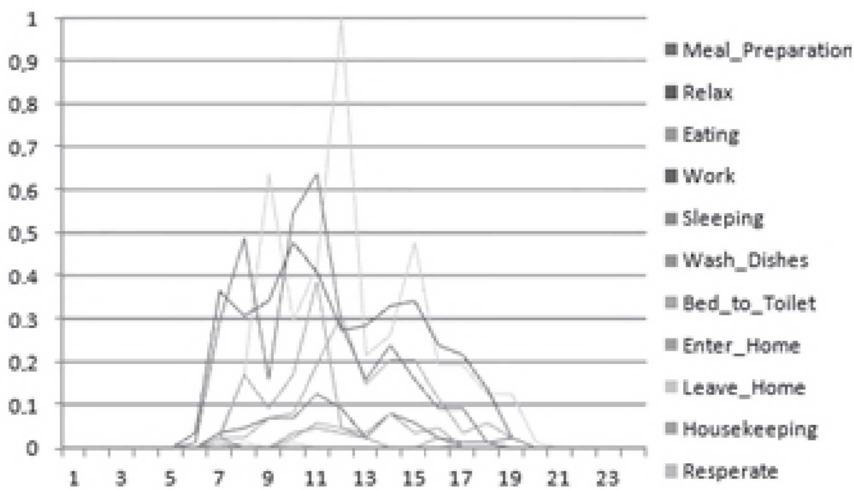

Fig. 3. Weights for the trained SHS-II model. The normalized weight values (y-axis) as a function of time quantized into chunks of one hour (x-axis)

Source: own elaboration.

As seen from the SHS-II trained weights, the correlation between the user activities can be found and thereby learned by SHS-II. This is also expected because most people have habits and follow the same system to some extent. Looking at the "leave home" curve it has peaks at 8, 11 and 15 hours, meaning that the user leaves the home most likely at these times. Focusing on "meal preparation" it can be seen that this peaks just before the "leave home" peaks. This means a correlation between preparing meals and afterwards leaving home properly exists. This is also what most people would be expected to do, for example, eat breakfast and then leave home for work. The other activities also seem to have the same kind of cross-correlation.

The question is then, whether the cases where the user leaves for more than one hour can be differentiated from the cases where the user will be back in one hour, based only on the activity weights. The performed research reveals that the above discussed correlation provides the necessary information. I.e. it is possible to detect the leave home situation with a good statistical probability.

The blue curve is the situations where the user leaves the home for more than one hour, whereas the red curve covers the situations where the user is away less than one hour. Comparing these curves it is obvious that some correlation exists between relaxing and leaving the home for more than one hour.

Thus, it can be observed that correlation is present, properly provided by the user habits, and that SHS-II is able to capture this correlation in its weights, that is, it learns.

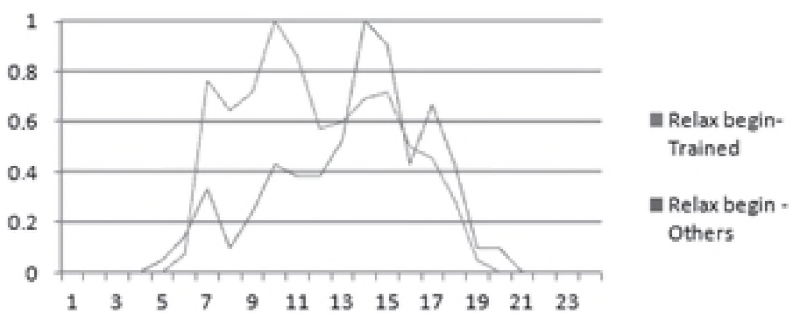

Fig. 4. The difference in activity relax begins when the user leaves the home for more than one hour (blue curve) and less than one hour (red curve)

Source: own elaboration.

As discussed, the SHS-II algorithm has been evaluated by using the leaving home scenario where the "leave home" activity is the predefined activity trigger type and the time differences between leaving home and entering home is more than one hour. Looking into the numbers the algorithm estimates the leaving home activity correctly approximately 75% of the time after learning from 295 user-annotated activity events. Thus, the estimation failure rate is 25% of the time. This is not a serious problem because from the user point of view, it means that an activity is not suggested to which the user response will probably be that the user performs this activity manually. However, this user interruption produces an annotated action-event from which the system learns and thereby improves, that is, it "bootstraps". More serious is the false positive (FP) outcome of 36%, because this means that the algorithm suggests an action that is not requested by the user's behaviour and which would probably be annoying for the user over time. This FP rate can be reduced by either adjusting the classifier parameters or by using more sensors to provide finer granularity of the information produced by the user's actions[10].

A direct comparison with other work is not simple, because, to the author's knowledge, the presented distributed smart home system has not been seen in this form before, that is, a combination of an action predictor using the Näive Bayes classifier that feeds an activity recognition system based on HMM has not been seen. However, it seems reasonable to compare SHS-II with other systems that predict user activity using a sequence of actions. One example is given in the work of Cheng-Tzong *et al.*[11] They designed a scenario based on a user activity prediction system named Adaptive Scenario Based Reasoning (ASBR) with a score of 80%. They also compared the performance of their system with a Case Based Reasoning (CBR) approach with a score of 75%. So, the results of the performance comparison is that they this presented system perform almost equally with other systems. However, it should be noted that these systems are quite different in their approaches and design even though the results are comparable.

Conclusions

A distributed smart home system has been presented. It offers a concept that combines a simple low level activity classifier named SHS-I with a high level one named SHS-II that is the target for this paper.

By using the public available CASAS data set it was found that the presented system behaves well compared to the CBR and ASBR systems. It achieves a true positive rate of 75% in the "leave home" scenario.

However, it should be noted that the threshold limit values are set manually, so further investigation is needed to clarify whether these limits are useable beyond the leave home scenario.

The future perspective of this work is to investigate the possibility of implementing SHS-II on different hardware platforms. Furthermore, an investigation of the look back depth in the SHS-II action buffer also needs investigation.

---

[10] Hongqing Fang, Raghavendiran Srinivasan, and Diane J Cook: *Feature Selections for Human Activity Recegnition*, International Journal of Innovative Computing, Information and Control, vol. 8, no. 5, 2012, pp. 3525–3535.

[11] Sheng Cheng-Tzong, Chi-Hsuan Wang, and Ching-Chung Chen: *An Adaptive Scenario Based Reasoning System cross smart houses*, Communications and Information Technologies, pp. 549–554, 2009.

Literature

1. Balasubramanian K. and Cellatoglu A.: *Improvements in home automation strategies for designing apparatus for efficient smart home*, Consumer Electronics, IEEE Transactions on, Vol. 54, No. 4, 2008.
2. Perumal T., Rmali A.R., and Chui Y.L.: *Interoperability Framework for Smart Home Systems*, IEEE Transactions on Consumer Electronics, Vol. 57, No. 4, 2011.
3. Starsinic M.: *System Architecture Challenges in the Home M2M Networks*, in Applications and Technology Conference (LISAT), Long Island Systems, 2010.
4. Bhardwaj S., Ozcelebi T., Lukkien J., and Uysal C.: *Resource and Service Management Architecture of a Low Capacity Network for Smart Spaces,* IEEE Transactions on Consumer Electronics, Vol. 58, No. 2, 2012.
5. Ye X., and Huang J.: *A Framework for Cloud-based Smart Home*, International Conference on Computer Science and Network Technology, 2011.
6. Kasteren T.V., Noulas A., Englebienne G., and Kröse B.: *Accurate Activity Recognition in a Home Setting,* UbiComp '08, September 21-24, Seoul, Korea., 2008.
7. Fang H., Srinivasan R., and Cook D.J.: *Feature Selections for Human Activity Recognition in Smart Home Environments*, International Journal of Innovative Computing, Information and Control, Vol. 8, No. 5, May 2012.
8. Cook D.J.: *Learning Setting-Generalized Activity Models for Smart Spaces*, Intellligent Systems IEEE, Vol. 27, No. 1, 2012.
9. Cook D.J.: *Learning Setting-Generalized Activity Models for Smart Spaces*, IEEE Intelligent Systems, Vol. 27, No. 1, 2012.
10. Fang H., Srinivasan R., and Cook D.J.: *Feature Selections for Human Activity Recegnition*, International Journal of Innovative Computing, Information and Control, Vol. 8, No. 5, 2012.
11. Sheng Ch.-T., Wang Ch.H., and Chen Ch.-Ch.: *An Adaptive Scenario Based Reasoning System cross smart houses*, Communications and Information Technologies, 2009.

DISTRIBUTED SMART HOME ACTIVITY RECOMMENDER SYSTEM USING HIDDEN MARKOV MODEL PRINCIPLES

Summary

A smart home is able to propose learned activities to its user and learn new activities by observing the user's behavioral patterns, that is, the user's actions. Most of today's discussed systems use some more or less complex classifier algorithms to predict user activities from contextual information provided by sensors. However, an alternative concept using a distributed framework is presented in this paper. It offers the possibility of combining simple low level activity classifiers with a more sophisticated one.

The high level classifier has been modeled in Java and tested on a publicly available data set that offers approximately seven months of annotated activity including 6468 sensor events produced by a women living in the test home. Using this data set, it has been shown that this system can achieve good performance with a recognition probability of 75%.

Translated by Per Lynggaard