Multi-resident Activity Recognition Using Incremental Decision Trees

Markus Prossegger[1] and Abdelhamid Bouchachia[2]

[1] Carinthia University of Applied Sciences,
Department of Engineering and IT, Klagenfurt, Austria
m.prossegger@cuas.at

[2] Bournemouth University, Faculty of Science and Technology,
Bournemouth, United Kingdom
abouchachia@bournemouth.ac.uk

Abstract. The present paper proposes the application of decision trees to model activities of daily living in a multi-resident context. An extension of ID5R, called *E-ID5R*, is proposed. It augments the leaf nodes and allows such nodes to be multi-labeled. E-ID5R induces a decision tree incrementally to accommodate new instances and new activities as they become available over time. To evaluate the proposed algorithm, the ARAS dataset which is a real-world multi-resident dataset stemming from two houses is used. E-ID5R performs differently on activities of both houses.

1 Introduction

Advances in ambient intelligence technology have become more prominent in the last decade yielding innovative and revolutionary applications related to smart environments such as smart homes, smart meeting rooms and classrooms, health monitoring and assistance systems, and smart factories. Usually smart environments aim at ensuring comfort, security, safety for the occupants and efficiency in the management of resources like energy. The technology of smart environments targets the design and development of smart adaptive systems capable of intelligently behaving by taking actions on behalf of the environments occupants for their satisfaction. In this setting, activity recognition plays an important role to achieve this capability, since perceiving and understanding the occupants behavior in the smart environment are crucial issues for the system to make a decision and to perform reasonable actions to the benefit of the occupant. Activity recognition is currently a challenging but exciting research topic because human activities are complex and are performed differently across individuals and become even more complex when the environment is inhabited by multiple occupants, which is the case in most real-world environments. The system needs to track each occupant when performing individual or group activities (e.g., move, seat, watch, garden, etc.) based on sensor readings with the overall goal to recognize what activity is being performed and to assist the occupant by taking actions on his/her behalf. For instance, in a smart home, the system should be

A. Bouchachia (Ed.): ICAIS 2014, LNAI 8779, pp. 182–191, 2014.

able to predict upcoming activities of each resident and send instructions to different smart devices to perform appropriate actions (such as starting a coffee machine once the occupant wakes up in the morning).

So far there has been a lot of effort for modeling human activities of daily living (ADL) in the context of pervasive computing and vision. In the former context, the smart homes are equipped with sensors and actuators, while in the later one, the living environment is equipped with cameras to capture data. Very often the use of cameras in criticized for privacy reasons. Using pervasive sensors allows to overcome the privacy issues and can be either installed in the environment or wearable by the resident.

Most of the published work related to ADL modeling has focused on environment occupied by one resident, called *single occupancy* environment. Many computational models have been used such as neural networks [2, 14, 15, 17], fuzzy rule-based systems [3–5, 9], decision trees [11, 20], hidden Markov models and similar graphical models [10, 12, 18, 19, 22].

So far, most smart home research has focused on monitoring and assisting single individuals in a single space. Since homes often have more than a single occupant, developing solutions for handling multiple individuals is vital. Dealing with multiple inhabitants has rarely been the central focus of research so far, as there have been numerous other challenges to overcome before the technology can effectively handle multiple residents in a single space. However, researchers are now beginning to recognize the importance of applying human activity recognition in smart homes with multiple inhabitants to design and develop real-world application needs.

Existing work on sensor-based activity recognition mainly focuses on recognizing activities of a single user [10, 12, 18, 19, 23]. Considering only single resident occupation is far from real life especially in smart homes. However modeling multiple-resident ADL is more complex because activities can take different forms: (1) Sequential activities, where each activity is performed after another, (2) Interleaving activities, where a single occupant switches back and forth between two or more activities, (3) Concurrent activities, where a single occupant performs two or more activities simultaneously, (4) Parallel activities, where occupants perform different activities , and (5) Collaborative activities, where the occupants work together in a cooperative manner where each occupant performs certain steps/actions of the activity, either together. Thus recognizing the individual occupants, known as the problem of data association and understanding their interaction with each other are key problems [24]. There has been some work on modeling and recognizing multi-resident activities based on pervasive sensors, but most of the work is based on computer vision [8]. The attempts made to model multi-resident activities have heavily relied on graphical models: the HMM [7], variants of HMM like Coupled Hidden Markov Model (CHMM) [6], Parallel Hidden Markov Model (PHMM) [6], Bayesian Networks (BN) [16], dynamic BN or variants of BN like DBN [13], Conditional Randoms Fields (CRF) or variants of CRF like Factorial Conditional Random Fields (FCRF) [25].

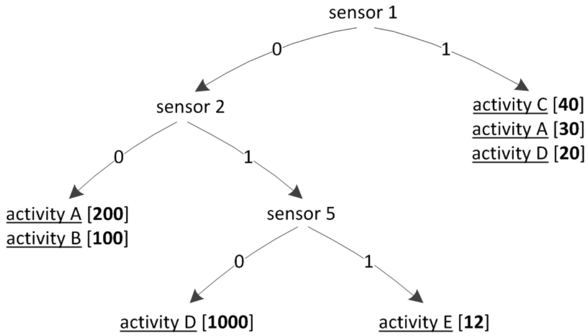

Fig. 1. Example of a decision tree. Each leaf node is multi-labeled (*underlined*) and each class is tagged with the number of occurrences (*bold*).

In this paper we will focus on multi-resident activity classification in the context pervasive computing. Instead of popular graphical models, we will use incremental decision trees (IDT). We propose an extension of ID5R [21], called *E-ID5R*. E-ID5R induces DTs with leaf nodes (class nodes) augmented by contextual information in the form of activity frequency.

The paper is organized as follows. Section 2 explains the process of decision tree induction using E-ID5R. Section 3 provides the experimental evaluation of the approach using ARAS data, Section 4 concludes the paper.

2 Details of E-ID5R

Decision trees are used in the context of this study to classify activities. Each activity is seen as a class and is described by a set of sensor readings. Unlike in conventional decision tree algorithms, in E-ID5R, the leaf nodes can represent single or multiple classes. Next to the class identifier, the number of the instances assigned to the class (class occurrence or class count) is encoded as a class attribute as shown in Fig. 1.

In order to construct a decision tree incrementally as new instances and activities become available, the following three stages are carried out: (1) Construction of an initial tree, (2) classification of new instances, and (3) evolution of the tree.

2.1 Construction of the Initial Tree

Based on a set of instances representing daily activities, where the input attributes are sensor measurements, an initial decision tree is generated using the steps described in Alg. 1. Because many activities can be conducted in the same space and are captured using the same set of sensors, ambiguity characterizes activities. Hence, the decision tree is adapted to accommodate multiple activities in a single leaf node. Figure 1 shows a simplified tree of input data representing

Algorithm 1. Decision tree generation

Require: training instances $i \in I$ with unique classification $c(i)$
for all $i \in I$ **do**
  if decision tree is empty $|N| == 0$ **then**
    Add single node n with all attribute-value pairs as unexpanded set.
    Add class $c(i)$ to classes C of node n.
    Set occurrence of class $c(i) = 1$.
  else
    Update the decision tree running Alg. 2.
  end if
end for

sensor measurements and the corresponding activities. Here the sensor measurements are given as binary values (1: emits data, 0: sensor is stale). To decide the activity corresponding to the input, the number of occurrences to the classes (class count) is added as class attribute.

Once the initial tree is constructed it can be used to classify new arriving unlabeled instances. In case the new instances are labeled, we can use them to evolve the initial tree.

2.2 Classification of New Instances

The tree can be used in a subsequent classification of new instances running Alg. 4. In the naive scenario, the number of occurrence encoded as class attribute is used as discriminator in case of a leaf node holding multiple classes.

As an example we examine the classification of a new instance giving a single measurement of *sensor 1 = 1* using the decision tree shown in Fig. 1: The leaf node holds the three candidate classes *activity C*, *activity A*, and *activity D* with the counts *40*, *30*, and *20*. Hence the candidate representing the highest count (*40*) is selected as the class of the new instance (*activity C*).

But if the instances are *time-stamped*, which represents the conventional scenario, the information can be used as discriminative attribute to further extend the approach. Next to the class count, the averaged time stamp (of all learned instances) is encoded as class attribute as shown in Fig.2. The weighted count Q^w of a class candidate $c \in C$ is calculated as follows:

$$Q^w(c) = \frac{Q(c)}{\sum_{i=1}^{C} Q(c_i)} + (1 - \frac{D(c)}{\sum_{i=1}^{C} D(c_i)})f \tag{1}$$

where C is the set of classes grouped in a single leaf and D is the distance between the time stamp of the instance to be tested $i \in I$, and the averaged time stamp of class c. The weighting of the distance can be tuned by the multiplicative factor f.

If the new instance is labeled with a unique time stamp of *50*, the weighted count of the classes change in the following way. Using Equation 1 and assuming $f = 1$, the count of class *activity C* changes from value 40 to round 0.80. The

Algorithm 2. Tree update

Require: instance i

repeat

  Find decision or leaf node n (with classes C) in tree, satisfying the tested attribute-value pairs of instance i.

  Update the classes counts at the tested attributes.

  Update the classes counts at the non-tested attributes.

  if n is a leaf node AND $c(i) \in C$ **then**

    Add non-tested attribute-value pairs of i as unexpanded set to n.

    Increase quantity of class $c(i)$.

  else if n is a leaf node AND $c(i) \notin C$ AND n does not contain non-tested attributes **then**

    Add class $c(i)$ to the classes C in leaf node n.

    Set quantity of class $c(i) = 1$.

  else if n is a leaf node AND $c(i) \notin C$ AND n contains non-tested attributes **then**

    Expand the non-tested attribute of n showing the highest information gain, creating one or more subtrees depending on the number of corresponding values.

    {Instance i will not be added in the current iteration}

  else

    Select one arbitrary non-tested attribute of decision node n and create a leaf node n_s.

    Assign $c(i)$ as class to leaf node n_s.

    Set quantity of class $c(i) = 1$.

    Add non-tested attribute-value pairs of i as unexpanded set to n_s.

  end if

until instance i was added

Transpose the tree following the predecessors of n respectively n_s running Alg. 3.

return update succeeded

count of *activity A* changes from 30 to 1.13 respectively from 20 to 1.08 in case of *activity D*. Considering the weighted count the new instance is labeled as class *activity A*.

2.3 Evolution of the Tree

Since our approach is based on an incremental decision tree induction algorithm, it can be evolved at any time using classified instances. This allows our approach to adapt to changes in the behavior of the residents over time. The instances can be more fine-grained and eventually new activities can be learned systematically. New arriving instances can be used to evolve the tree by running Alg. 2. The quality of the classification using the proposed approach is shown in the experimental section.

Algorithm 3. Transpose tree

Require: leaf node n
{Traverse the tree bottom-up starting with n}
while there is a predecessor n_p of n **do**
  if $information\_gain(n) > information\_gain(n_p)$ **then**
    Swap the places of n_p and n.
    Reorder all other subtrees of n_p if there are any.
  end if
  Select n_p as n during the next iteration.
  {Do one step bottom-up}
end while

Algorithm 4. Classification

Require: new unlabeled instance i
Find leaf node n (with classes C) in the tree, satisfying the tested attribute-value pairs of instance i.
if $|C| == 1$ **then**
  return i is an instance of class C.
else if $|C| > 1$ **then**
  Select class $c \in C$ having the highest value $Q(c)$.
  return i is an instance of class c.
else
  return i is an instance of an unknown class.
end if

3 Evaluation

To evaluate E-ID5R we use the dataset ARAS (Activity Recognition with ambient sensing) which is known to present a multi-user setting. It contains information about the association (activity, resident), that is which resident performs which activity. Many activities are either parallel or cooperative ones. The dataset covers a full month of labelled activities for multiple residents in two real houses. A total number of 27 different activities is described by the binary measurement force sensitive resistors, pressure mats, contact sensors, proximity sensors, sonar distance sensors, photocells, temperature sensors, and infra-red receivers. Each instance from *House A* or *House B* represents a unique activity (i.e., combination of the residents' activities) combined with the sensor measurements at a given time stamp. The instances from day 1 to day 21 were used to generate and evolve the decision tree. The instances from day 22 to day 28 were used to measure the classification accuracy using the generated tree. Each of the data sets *House A* and *House B* contains $1,814,400$ (day 1..21) learning instances used to construct the decision tree and $604,800$ (day 22..28) test instances to evaluate the classification. The description of the training nad testing data used are shown in Table 1 and Table 2 respectively.

An example of an ambiguous combination is interpreted the following: The sensors' readings (in *House A*) for the activity *Resident 1 is talking on the phone*

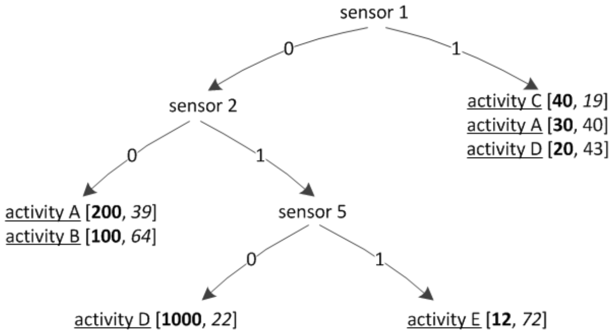

Fig. 2. Example of an enhanced decision tree. Each class is extended with a count (*bold*) and an averaged time stamp (*italic*) of occurrence.

Table 1. Description of the learning data

ID	Number of different classes	Number of ambiguous combinations	Number of ambiguous measurements
House A	243	24,236	3,200 (of total 3,732)
House B	147	6,352	700 (of total 879)

& resident 2 is using the Internet and the activity *Resident 1 is watching TV & resident 2 is using the Internet* are the same. An ambiguous measurement indicates two (or more) activities have the same sensor reading.

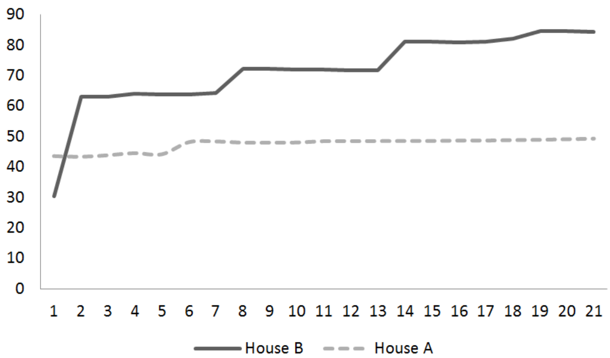

Fig. 3. Percentage of correct classification with respect to the number of days provided as learning data

In Table 3 the result of a number of selected evaluation experiments is shown. Figure 3 shows the number of correct classified instances with respect to the amount of learning data. Although the percentage of correct classified instances

Table 2. Dimension of the test data

ID	Number of different classes	ambiguous combinations	ambiguous measurements
House A	177	11,129	1,660 (of total 2,065)
House B	108	3,768	355 (of total 464)

Table 3. Selected results

ID	Learning instances [day]	Test instances [day]	Correct [%]	Incorrect due to count [%]	no/wrong leaf [%]
House A	1..1	22..28	43.56	15.06	41.38
	1..7	22..28	48.36	30.91	20.73
	1..14	22..28	48.53	38.30	13.17
	1..21	22..28	49.28	40.31	10.41
House B	1..1	22..28	30.47	1.06	68.47
	1..7	22..28	64.19	6.00	29.81
	1..14	22..28	81.08	9.26	9.66
	1..21	22..28	84.45	10.35	5.2

seems to be low, it increases steadily with the number of instances to be learned. The extended approach considering the weighted count as described in Equation 1 improves the percentage of correctly classified instances by an average of 1% only. The performance is very prone to the selected value of factor f and could be improved if an heuristics can be found to predetermine it efficiently.

4 Conclusion

In this paper, an algorithm, called E-ID5R, to induce incremental decision trees is proposed to deal with the classification of multi-resident activities. Initial experiments show that the prediction of activities for House A presents a quite challenging task. The classification rate is insignificantly as much low as 40%. Clearly the parallel and cooperative activities need a better modeling than straight and naive application of decision trees. In the case of House B, the results are much better approaching 82% when the first half duration is used for training and the second is used for testing.

Considering the outcome of the set of experiments, the efficiency of multi-labeling and the use of counts must be further analysed, especially the effect of value of factor f in Equation 1.

References

1. Alemdar, H., Ertan, H., Incel, O.D., Ersoy, C.: ARAS Human Activity Datasets in Multiple Homes with Multiple Residents. In: 2013 7th International Conference on Proceedings Pervasive Computing Technologies for Healthcare (PervasiveHealth), pp. 232–235 (2013)

2. Begg, R., Hassan, R.: Artificial Neural Networks in Smart Homes. In: Augusto, J.C., Nugent, C.D. (eds.) Designing Smart Homes. LNCS (LNAI), vol. 4008, pp. 146–164. Springer, Heidelberg (2006)
3. Bouchachia, A., Vanaret, C.: GT2FC: An Online Growing Interval Type-2 Self-Learning Fuzzy Classifier. IEEE Transactions on Fuzzy Systems (in Press, 2014), doi:10.1109/TFUZZ.2013.2279554
4. Bouchachia, A.: Fuzzy classification in Dynamic Environments. Soft Computing 15(5), 1009–1022 (2011)
5. Bouchachia, A.: An Evolving Classification Cascade with Self-Learning. Evolving Systems 1(3), 143–160 (2010)
6. Chiang, Y., Hsu, K., Lu, C., Fu, L., Hsu, J.Y.: Interaction Models for Multiple-Resident Activity Recognition in a Smart Home. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3753–3758 (2010)
7. Cook, D., Crandall, A., Singla, G., Thomas, B.: Detection of Social Interaction in Smart Spaces. Journal of Cybernetics and Systems 41, 90–104 (2010)
8. Du, Y., Chen, F., Xu, W., Li, Y.: Recognizing Interaction Activities using Dynamic Bayesian Network. In: Proceeings of the 18th International Conference on Pattern Recognition, pp. 618–621 (2006)
9. Hagras, H., Doctor, F., Callaghan, V., Lopez, A.: An Incremental Adaptive Life Long Learning Approach for Type-2 Fuzzy Embedded Agents in Ambient Intelligent Environments. IEEE Transactions on Fuzzy Systems 15(1), 41–55 (2007)
10. Hu, D., Zhang, X., Yin, J., Zheng, V., Yang, Q.: Abnormal Activity Recognition based on HDP-HMM Models. In: Proceedings of IJCAI 2009, pp. 1715–1720 (2009)
11. Isoda, Y., Kurakake, S., Nakano, H.: Ubiquitous Sensors based Human Behavior Modeling and Recognition using a Spatio-Temporal Representation of User States. In: Proceedings of the 18th International Conference on Advanced Information Networking and Applications, pp. 512–517 (2004)
12. Khan, S., Karg, M., Hoey, J., Kuli, D.: Towards the Detection of Unusual Temporal Events during Activities Using HMMs. In: Proceedings of UbiComp 2012 (2012)
13. McCowan, I., Gatica-Perez, D., Bengio, S., Lathoud, G., Barnard, M., Zhang, D.: Automatic Analysis of Multimodal Group Actions in Meetings. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(3), 305–317 (2005)
14. Mozer, M.: The Neural Network House: An Environment that Adapts to its Inhabitants. In: Proceedings of the American Association for Artificial Intelligence Spring Symposium on Intelligent Environments, pp. 110–114 (1998)
15. Li, H., Zhang, Q., Duan, P.: A Novel One-pass Neural Network Approach for Activities Recognition in Intelligent Environments. In: Proceedings of the 7th World Congress on Intelligent Control and Automation, pp. 50–54 (2008)
16. Lin, Z., Fu, L.: Multi-user Preference Model and Service Provision in a Smart Home Environment. In: Proceedings of IEEE International Conference on Automation Science and Engineering, pp. 759–764 (2007)
17. Lotfi, A., Langensiepen, C., Mahmoud, S., Akhlaghinia, M.: Smart Homes for the Elderly Dementia Sufferers: Identification and Prediction of Abnormal Behaviour. Journal of Ambient Intelligence and Humanized Computing 3(3), 205–218 (2012)
18. Pérez, Ó., Piccardi, M., García, J., Patricio, M.A., Molina, J.M.: Comparison Between Genetic Algorithms and the Baum-Welch Algorithm in Learning HMMs for Human Activity Classification. In: Giacobini, M. (ed.) EvoWorkshops 2007. LNCS, vol. 4448, pp. 399–406. Springer, Heidelberg (2007)

19. Sarkar, J., Lee, Y., Lee, S.: ARHMAM: an Activity Recognition System based on Hidden Markov mined Activity Model. In: Proceedings of the 4th International Conference on Ubiquitous Information Management and Communication (ICUIMC 2010), pp. 484-492 (2010)
20. Stankovski, V., Trnkoczy, J.: Application of Decision Trees to Smart Homes. In: Augusto, J.C., Nugent, C.D. (eds.) Designing Smart Homes. LNCS (LNAI), vol. 4008, pp. 132–145. Springer, Heidelberg (2006)
21. Utgoff, P.: Incremental Induction of Decision Trees. Machine Learning 4, 161–186 (1989)
22. van Kasteren, T., Noulas, A., Englebienne, G., Kroese, B.: Accurate Activity Recognition in a Home Setting. In: Proceedings of UbiComp 2008, pp. 1–9. ACM (2008)
23. van Kasteren, T.L.M., Englebienne, G., Kröse, B.J.A.: Hierarchical Activity Recognition Using Automatically Clustered Actions. In: Keyson, D.V., et al. (eds.) AmI 2011. LNCS, vol. 7040, pp. 82–91. Springer, Heidelberg (2011)
24. Wang, L., Gu, T., Tao, X., Lu, J.: Sensor-Based Human Activity Recognition in a Multi-user Scenario. In: Tscheligi, M., de Ruyter, B., Markopoulus, P., Wichert, R., Mirlacher, T., Meschterjakov, A., Reitberger, W. (eds.) AmI 2009. LNCS, vol. 5859, pp. 78–87. Springer, Heidelberg (2009)
25. Wang, L., Gu, T., Tao, X., Chen, H., Lu, J.: Recognizing Multi-user Activities using Wearable Sensors in a Smart Home. Pervasive and Mobile Computing 7(3), 287–298 (2011)