Multimedia Tools and Applications (2020) 79:23445–23460
https://doi.org/10.1007/s11042-020-09093-0

Mixed-dependency models for multi-resident activity recognition in smart homes

Son N. Tran[1] · Tung-Son Ngo[2] · Qing Zhang[3] · Mohan Karunanithi[3]

Received: 20 March 2019 / Revised: 8 April 2020 / Accepted: 22 May 2020 /
Published online: 9 June 2020

Abstract

Recent growing interest in ambient intelligent environments has driven a desire for effective models to reason about activities of multiple residents. Such models are the keystone for the future of smart homes where occupants can be assisted with non-intrusive technologies. Much attention has been put on this research, however current works tend to focus on developing statistical algorithms for prediction, whilst there still lacks a study to fully understand the relations of residents' behaviours and how they are reflected through the sensors' states. In this paper we investigate the dependencies of the activities from residents and their interaction with the environments. We represent such dependencies in Bayesian networks that leads to construction of six variants of Hidden Markov Models (HMMs). Furthermore, we argue that a complete model should embody more than one type of dependency. Therefore, we propose an ensemble of HMMs, and then generalize it to a novel mixed-dependency model. In the experiments we perform intensive evaluation of our study on multi-resident activity recognition task. The results show that the proposed models outperform other models in three smart home environments, thus asserting our hypothesis.

Keywords Smart homes · Multi-resident activity recognition · Hidden Markov Models

1 Introduction

Activity monitoring in ambient smart homes has been attracting a great deal of attention recently due to its promising application in aged care where privacy and obtrusiveness are

✉ Son N. Tran
sn.tran@utas.edu.au

[1] The Discipline of ICT, University of Tasmania, Newnham Campus, Launceston, Australia

[2] Department of Computer Science, FPT University, Hanoi, Vietnam

[3] The Australian E-Health Research Centre, CSIRO, Royal Brisbane and Women's Hospital, Level 5, UQ Health Science building, Brisbane, Australia

the major concerns [3]. Different from visual-based approaches that employ video data for learning and reasoning about the activities of multiple residents, ambient approaches rely on low-cost, power-saving sensors placed in different locations in a smart home. Although the use of cameras can provide rich information for accurately modelling human behaviour, many users feel uncomfortable with the appearance of cameras and also fear that their identities and sensitive activities can be unnecessarily revealed. By contrast, ambient sensors are only triggered when a resident performs an activity and therefore being less intrusive. Wearable sensors, perhaps, are currently the most popular and successful tool for personal activity tracking. However, in many cases, especially in aged care for multiple occupants, this approach is expensive and is not always welcomed by senior citizens who are not comfortable with having an electronic device attached. Such limitations have motivated research on the future of smart home where human activities and interactions can be measured through sparse, indicative sensor data from ambient devices.

Research on activity monitoring in ambient environment have been active in the past fifteen years with the focus is shifting from static approaches such as k-Nearest Neibourgh, Decision Trees, Multi-layer Perceptron [12, 29], incremental Decision Tree (IDT) [23] to temporal approaches such as Hidden Markov Models (HMM) [1, 5, 8, 11, 25], Conditional Random Field (CRF) [3, 10, 18, 30], and Recurrent Neural Networks [28]. The central idea of these approaches is to employ a data-driven model to learn a classifier for activity recognition with inputs are sensors' values. Data-driven approaches have also been used to predict the time when an activity will occur which would be useful for early prevention and anomaly detection [20, 21]. With the success of deep learning recently, some sequence modelling approaches have been adopted for activity recognition with ambient sensor data, most of them are inspired by natural language processing. For example, in [28] the recognition module of the Smarter Safer Home platform is built upon Gated Recurrent Unit [7] and Long Short Term Memory [17]. In [31] the authors take the advantage of stacked Auto-encoders in learning high-level, predictive features for activity recognition with binary sensor data. In [16],sequence-to-sequence model [26] are used to learn temporal features. Besides investigating new activity modelling approaches, a heterogeneous strategy was also used to transfer activity information from one sensor platform to another to boost the performance [14].

As we can see, most of the research currently focuses on improving the prediction performance, while in a multi-resident environment, there is an unanswered question of whether the activities of a resident can influence the others. In [2], the activities are segmented to detect activity transitions. However, this work does not show how the activities of residents are related. We argue that the activity of a resident at a time step can be seen as a frequent pattern, like a habit (having breakfast after getting up in the morning), or as a part of a collaborative event (one is cooking and the other is cleaning the dining table). This seems that the activity of a resident depends on the previous activities and also from the activities of other residents. In this paper, instead of developing a new model for performance improvement, we are interested in study the relationship of activities in multi-resident smart home environments.

In the first part of this paper we show an investigation into the behaviour dependencies in smart home environments. We classify the dependencies of activities as: *individual dependency*: Activities of a resident depend only on his previous activity; *cross dependency*: Activities of a resident depends on the previous activities of all residents; and *group dependency*: Activities of all residents depends on their previous activities. We also consider two forms of interactions between activities and environment's states. Here the states are triggered either by each individual independently or by the activities of all residents at a time.

We model the dependencies into six variants of Hidden Markov Models (HMMs), some of which have not been used for multi-resident activity recognition in ambient environments before.

The second part of this paper is a proposal of a novel approach for multi-resident modelling. We argue that a complete model should embody more than one type of dependency to be able to present the complexity of collaborative, interactive activities. We then propose an ensemble of HMMs, which we call as multi-dependencies HMM (md-HMM), to implement the idea. A md-HMM is a combination of several HMMs having different transition probability tables while sharing the same emission probability table. It is obvious that the role of each type of dependency varies in different environments, depending on the living styles of occupants. Therefore, we further generalize the ensemble to a mixed dependencies model (MDM). Different from md-HMM where the model's log-probability is the sum of the other HMMs, MDM is represented by a weighted log-probability. At a special case, a MDM can be seen very similar as md-HMM.

We carry out experiments on three smart home environments from CASAS [9, 18] and ARAS [1] datasets. As far as we know, this is the first work that performs intensive empirical evaluation using multiple smart home environments with different types of feature representation. For reproducibility we share our code at https://github.com/sFunzi/mdm. Among six variants of HMMs we find that the HMMs with *group dependency* is more accurate than the other variants of HMMs. We also find that representing the environment's state as a result of all residents' activities is better than separate it for each individual. More importantly, the empirical results confirm our hypothesis in which mixed dependencies indeed capture the complexity of multi-resident activities. In particular, the md-HMM has better performance than other models and MDM can even achieve further improvement.

The organization of this paper is as follows. In the next section, we discuss the dependencies in ambient smart home environments with multiple occupants. Section 3 shows how to combine such dependencies to construct different HMMs for activity modelling. In Section 4 a ensemble of HMMs, called md-HMM, and its generalisation MDM are proposed. Related work is presented in Section 5. We showcase the empirical study in Section 6 and perform intensive experiments on three smart home environments from two datasets. The last section concludes the work.

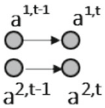
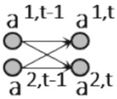
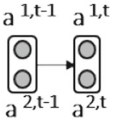
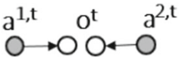
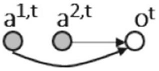
2 Smart home environments

In a seamless smart homes we can utilise ambient devices to monitor the behaviour of residents by attaching them to various locations. Such devices such as motion and force sensors are affordable with low-energy consumption which are very suitable for mass production in future.

2.1 Notations

Let us denote $a^{m,t}$ and o^t as an activity of resident m and sensors' state at time t respectively. For ease of presentation we denote $\mathbf{a}^t = \{a^{1,t}, a^{2,t}, .., a^{M,t}\}$ as activities of all M residents at time t. We use $t_1 : t_2$ to denote a sequence of events/states from time t_1 to t_2. For example, $\mathbf{a}^{t_1:t_2} = \{\mathbf{a}^{t_1}, .., \mathbf{a}^{t_2}\}$ is a sequence of activities performed by all residents from time t_1 to t_2.

Table 1 First row (left to right): parallel dependency, cross dependency, group dependency ; First column (top to bottom): individual interaction, group interaction; pHMM: parallel HMM, cHMM: coupled HMM, gd-cHMM: coupled HMM with group dependency, fHMM: factorial HMM, cd-fHMM: factorial HMM with cross dependency, gd-HMM: HMM with group dependency

			
	pHMM	cHMM	gd-cHMM
	fHMM	cd-fHMM	gd-HMM

2.2 Activity dependencies

For multiple residents there are three types of dependencies as being illustrated in the first row of Table 1. Let us take two residents as an example. First, each resident's activities are seen as an independent Markov chain where there is no interactive link between them. We call this *parallel dependency*, as shown in the left figure . Second, we assume that activity of a resident at time t depends not only on his previous activity but also on the activities of the others. This seems to make sense since in smart homes occupants tend to interact with others. We call this *cross dependency*, as shown in the middle figure. Finally, we can treat the group of activities of all residents as a single random variable where their current combined activity depends on the previous one, as shown in the right figure. We call this *group dependency*.

2.3 Environment interaction

We now consider two types of interactions between activities and environment's states, as can be seen in the first column of Table 1. The states of a smart home are normally represented by the values of sensors, denoted as o^t at time t. A state is recorded when one or more residents perform some activities and therefore trigger the sensors. In the first case, as shown in the top figure, each resident has his own interaction with the environment. This based on the fact that a sensor can only be triggered by a single person. Note that in non-intrusive ambient environments, it is difficult to associate the activated sensors to the person who activates them. Instead, states of environment are replicated for each person. In the second case, one may argue that since the environment is treated as one object, its states should reflect the whole dynamic in it. Therefore, in this case the environment's state is modelled to be dependent on the activities of all residents, as shown in the bottom figure.

3 Multi-resident activities modelling

In this section we show how to model the activities of multiple residents in smart homes by combining the dependencies discussed in the previous section.

3.1 Hidden Markov models

A HMM [24] consists of a single hidden y and an observation variable x which assumes a Markov process. It represents a state sequence as a joint distribution as:

$$p(y^{1:T}, x^{1:T}) = p(x^1|y^1)p(y^1)\prod_{t=2}^{T} p(x^t|y^t)p(y^t|y^{t-1}) \tag{1}$$

This is parameterised by the probability tables $p(x^t = i|y^t = j)$, $p(y^t = j|y^{t-1} = j')$, $p(y^1 = j)$, which are called emission, transition, and prior probabilities respectively. In order to learn the model's parameters, one would like to maximize the log-likelihood: $\ell = \sum_{y^{1:T}, x^{1:T} \in \mathcal{D}} \log(p(y^{1:T}, x^{1:T})$, where $y^{1:T}, x^{1:T} \in \mathcal{D}$ is a sequence of inputs and output, e.g sensors' states and activities, in the training data $\mathcal{D}$. Given a new sequence of observations, prediction can be performed by finding the most probable hidden states using Viterbi algorithm.

3.2 Multi-resident activity models

HMM is perfectly useful for modelling the dependencies we have discussed in Section 2. The activities of residents can be seen as multiple hidden variables while the environment's states can be either presented as a single variable or as multiple replicas of a variable. By considering each type of dependencies and interactions we come up with six variants of HMM as shown in Table 1. In what follows we discuss each of them in detail.

3.2.1 pHMM

We can model each resident's activities by a separated HMM, similar to the model proposed in [6]. However, it should be noted that in that work the data association is provided such that the input of each HMM is tied with only the states of the sensors triggered by a specific person. In the general case as we are studying in this paper such information is not available. Therefore the input for each HMM should be replicated for all residents. The joint distribution of this parallel HMM is:[1]

$$p = \prod_{m} \left[p(o^1|a^{m,1})p(a^{m,1}) \prod_{t=2}^{T} p(o^t|a^{m,t})p(a^{m,t}|a^{m,t-1}) \right] \tag{2}$$

The parameters in this model is different from those in the single HMM where we only need three probability tables. Here, the model would have M transition probability tables, M emission probability tables, and M priors. Each HMM will be learned independently using the same algorithm. For prediction, each HMM will predict the activities of a resident and the results are combined from all HMMs for evaluation.

3.2.2 cHMM

Parallel HMM has an issue in that it does not take into account the relations of residents' activities. Each HMM assumes that the current activity of a resident depends only on his previous activity. As we mentioned earlier, this may not reflect the real situation in smart

[1] Here p denotes $p(\mathbf{a}^{1:T}, o^{1:T})$ to save the presentation space.

homes where activities of a resident depend on other residents at some time. Therefore, by coupling the hidden variables of separate HMMs while maintaining the replication of observation variables we can have a new model that capture such cross dependency, similar as in [6]. In this case the joint distribution is:

$$p = \prod_{m} \left[p(o^1|a^{m,1}) p(a^{m,1}) \prod_{t=2}^{T} p(o^t|a^{m,t}) p(a^{m,t}|\mathbf{a}^{t-1}) \right] \tag{3}$$

Here, the emission probabilities are the same as those in the parallel HMM but the transition probabilities are different. We also use Viterbi algorithm to infer the most probable activities given a sequence of sensors' states. Due to the coupling, we are not able to perform parallel inference as in the previous model. Instead, we apply the Viterbi algorithm by replacing $p(x^t|y^t)$ with $\prod_m p(o^t|a^{m,t})$, $p(y^t|y^{t-1})$ with $\prod_m p(a^{m,t}|\mathbf{a}^{t-1})$, and $p(y^1)$ with $\prod_m p(a^{m,1})$.

3.2.3 gd-cHMM

A gd-cHMM has similar structure as the coupled HMM, the only difference is that the hidden variables are coupled by group dependency instead of cross dependency.

$$p = \prod_{m} p(o^1|a^{m,1}) p(\mathbf{a}^1) \prod_{t=2}^{T} \prod_{m} p(o^t|a^{m,t}) p(\mathbf{a}^t|\mathbf{a}^{t-1}) \Bigg] \tag{4}$$

Since the same environment dependency is used, this model has the same emission probabilities as pHMM and cHMM. The transition table in this case should have higher storage complexity than two previous cases. For prediction, we replace $p(x^t|y^t)$ with $\prod_m p(o^t|a^{m,t})$, $p(y^t|y^{t-1})$ with $p(\mathbf{a}^t|\mathbf{a}^{t-1})$, and $p(y^1)$ with $p(\mathbf{a}^1)$ before applying the Viterbi algorithm.

3.2.4 fHMM

Factorial HMM was proposed by Ghahramani and Jordan in [15]. This can be seen as a generalization of HMMs where the single hidden variable is factored into multiple hidden variables. To apply the model to multi-resident activity recognition, we assign each hidden variable to represent a resident's activities. One can see it as similar as the parallel HMM except that there is only a single observation. We take into account that in pHMM and cHMM the sensors depend on each individual's activities separately, which is only valid when the data association is available. Without this, separating the observation of each resident may lead to drop in performance as what we will show in the experiments. Therefore, factorial HMM has one single observation variable, hopefully to solve such problem. The joint probability of the fHMM is:

$$p = p(o^1|\mathbf{a}^1) \prod_{m} p(a^{m,1}) \prod_{t=2}^{T} \big[p(o^t|\mathbf{a}^t) \prod_{m} p(a^{m,t}|a^{m,t-1}) \Bigg] \tag{5}$$

Similar to other HHM-based models this factorial HMM will be learned by maximizing the log-likelihood. Once the parameters are learned, we can use the model to perform prediction task through Viterbi algorithm, as in Section 3.1. In this case we just need to replace $p(x^t|y^t)$ by $p(o^t|\mathbf{a}^t)$, $p(y^t|y^{t-1})$ by $\prod_m p(a^{m,t}|a^{m,t-1})$ and $p(y^1)$ by $\prod_m p(a^{m,1})$

3.2.5 cd-fHMM

In order to represent the relations between activities among residents as what has been discussed in Section 2, we add cross connections from all hidden variables at time $t-1$ to each hidden variable at time t. This results in a fHMM model with cross dependency. In this model, the joint probability of sensors' states and activities of all residents is:

$$p = p(o^1|\mathbf{a}^1)\prod_m p(a^{m,1})\prod_{t=2}^{T}(p(o^t|\mathbf{a}^t)\prod_m p(a^{m,t}|\mathbf{a}^{t-1})) \tag{6}$$

It can be seen that only the transition probabilities are changed in comparison to the fHMM above. For inference, similar to the fHMM we can apply the Viterbi algorithm with substitutions of $\prod_m p(a^{m,t}|\mathbf{a}^{t-1})$ and $p(\mathbf{a}^1)$ for $p(\mathbf{a}^t|\mathbf{a}^{t-1})$ and $\prod_m p(a^{m,1})$ respectively.

3.2.6 gd-HMM

The last variant we study in this paper is the HMM with group dependency which can be seen as a single HMM with one hidden variable to represent the combined activities of all residents. The joint distribution of this HMM for multi-resident activity modelling simply is:

$$p = p(o^1|\mathbf{a}^1)p(\mathbf{a}^1)\prod_{t=2}^{T} p(o^t|\mathbf{a}^t)p(\mathbf{a}^t|\mathbf{a}^{t-1}) \tag{7}$$

Compare to the other variants this model require larger storage for emission probability table, similar as gd-cHMM. However, this may be useful for inference since it does not need to combine M small transition probability tables as in the other HMMs, except pHMM.

4 Mixed-dependency models

Previous section studies various variants of HMMs, each represents a type of activity dependency and interaction in smart home environments. We argue that the complexity of multi-resident activities would require more than one type of dependency for better reasoning. In this section, first we propose an ensemble of HMMs to combine different type of activity dependencies. Then we generalize the idea to propose another novel model that mixes the dependencies and subsumes the ensemble.

4.1 Ensemble model

Let us consider an ensemble of fHMM, cd-HMM and gd-HMM where parallel dependency, cross dependency and group dependency are combined. Note that for simplicity we only use the HMMs that have the same representation of environment. The idea here is to constrain the HMMs together such that the most likely sequence of activities must maximise the combined probabilities of all HMMs. For example, we can represent the ensemble as the sum of the probabilities as: $p_{\text{gd-hmm}} + p_{\text{cd-hmm}} + p_{\text{fhmm}}$. With this, the learning is efficient by applying maximum likelihood estimate to each model separately. However, we are not sure that whether dynamic programming algorithm in HMMs, i.e. Viterbi, can be applicable to the sum of probabilities. Therefore, to ease the inference we propose an ensemble model which is formularised in a closed form of the combined probabilities in log-space as:

$$\phi_{\text{md-HMM}} = \log p_{\text{gd-hmm}} + \log p_{\text{cd-hmm}} + \log p_{\text{fhmm}} \tag{8}$$

We call this ensemble as mix-dependency HMM or md-HMM. After training the md-HMM by maximising the log-likelihood of each HMM in the ensemble we can combine them for prediction as:

$$\mathbf{a}^{*1:T} = \arg\max_{\mathbf{a}^{1:T}} (\phi_{\text{md-HMM}}(o^{1:T}, \mathbf{a}^{1:T})) \tag{9}$$

This can be done through dynamic programming, similar as in HMMs. Let us denote $\mu_t = \max_{\mathbf{a}^{1:t-1}} p(\mathbf{a}^t = j, \mathbf{a}^{1:t-1}, o^{1:t})$ we have:

$$\begin{aligned}\mu_t(j) = {} & \log(p(o^t|\mathbf{a}^t = j)) + \max_{j'}[\log(p_{\text{gd-hmm}}(\mathbf{a}^t = j|\mathbf{a}^{t-1} = j')) \\ & + \log(p_{\text{cd-hmm}}(\mathbf{a}^t = j|\mathbf{a}^{t-1} = j')) + \log(p_{\text{fhmm}}(\mathbf{a}^t = j|\mathbf{a}^{t-1} = j')) \\ & + \mu_{t-1}(j')]\end{aligned} \tag{10}$$

In order to find the most probable activities, first we find $\mathbf{a}^{*T} = j^* = \arg\max_j \mu_T(j)$ and then trace back to get $\mathbf{a}^{*T-1} = \arg\max_{j'}[\log(p_{\text{gd-hmm}}(\mathbf{a}^T = j^*|\mathbf{a}^{T-1} = j')) + \log(p_{\text{cd-hmm}}(\mathbf{a}^t = j|\mathbf{a}^{t-1} = j')) + \log(p_{\text{fhmm}}(\mathbf{a}^T = j^*|\mathbf{a}^{T-1} = j')) + \mu_{T-1}(j')]$ in (10). We repeat this process to infer the whole sequence of activities $\mathbf{a}^{*T}$, $\mathbf{a}^{*T-1}$, ..., $\mathbf{a}^{*1}$, which can be done efficiently using dynamic programming.

4.2 Mixture of dependencies

We observe that the emission probability table does not have important role as the transmission probabilities in activity modelling. We also find that the influence of each type of dependency varies in different environments, depending on the complexity of the occupants' activities. Therefore we generalize the log-probability in the ensemble such that each type of dependency is assigned with a different weights. This can be seen as a mixture of weighted log-probabilities which we call mixed-dependency model (MDM). MDM is a single model which capture all the dependencies, rather than an ensemble of different HMMs. The combined log-probability of this model is:

$$\begin{aligned}\phi_{\text{MDM}} = {} & \log p(o^t|\mathbf{a}^t) + \alpha \log p(\mathbf{a}^0) + (\beta + \gamma) \sum_m \log p(a^{m,0}) \\ & + \sum_t \left[\alpha \log p(\mathbf{a}^t|\mathbf{a}^{t-1}) + \sum_m \beta \log p(a^{m,t}|\mathbf{a}^{t-1}) + \gamma \log p(a^{m,t}|a^{m,t-1}) \right]\end{aligned} \tag{11}$$

where α, β, γ are non-negative weights. We can show that this MDM subsumes fHMM, cd-fHMM, gd-HMM and also the ensemble md-HMM. Indeed, the combined log-probability of MDM is equivalent to the log-probabilities of fHMM, cd-fHMM and gd-HMM with the assignments ($\alpha = 0, \beta = 0, \gamma = 1$), ($\alpha = 0, \beta = 1, \gamma = 0$) and ($\alpha = 1, \beta = 0, \gamma = 0$) respectively. Similarly, if we set $\alpha = \beta = \gamma = 1/3$ we would have $\phi_{\text{MDM}} \propto \phi_{\text{md-HMM}}$. Interestingly, if we maximise the combined log-likelihood from ϕ_{MDM} given a constraint that $\alpha + \beta + \gamma = 1$ we end up in finding the best model among fHMM, cd-fHMM and gd-HMM. This is similar as applying linear programming to maximise the log-likelihood of MDM while setting α, β, γ as variables. In practice, these values are selected empirically as shown in the experiments.

Similar to (10) in md-HMM, inference in MDM is efficient with $\mu_t(j)$ as:

$$\mu_t(j) = \log(p(o^t|\mathbf{a}^t = j)) + \max_{j'}[\alpha \log p(\mathbf{a}^t|\mathbf{a}^{t-1}) + \sum_m \beta \log p(a^{m,t}|\mathbf{a}^{t-1}) + \gamma \log p(a^{m,t}|a^{m,t-1}) + \mu_{t-1}(j')] \quad (12)$$

5 Related work

In multi-resident smart homes, HMMs [24] have been studied intensively, as being showed in previous studies [1, 5, 8, 25]. The first model could be employed is single HMM. However, due to the complexity of multiple activities it may need some modification. For example, the activities can be combined as joint labels so that they can be represented by a single hidden variable [5]. Another method to model the activities of multiple residents is to create multiple HMMs, one for each resident [6]. Such model, as known as parallel HMM, has been evaluated in the case that data association is provided. Another approach is coupling HMMs by assuming that the activity of a resident is dependent not only on his previous activity but also on the previous activities of other residents. There were proposals of coupled HMM and factorial HMM in computer vision domain [4], but only cHMM was employed for sensor data [6]. Combining HMMs of the same type was studied in [13]. Different from that, in this paper we ensemble HMMs of various types for activity recognition. Besides HMMs, Conditional Random Fields (CRFs) [10, 18] and incremental decision trees (IDT) [23] also have been used for multi-resident activity recognition.

6 Experiments

6.1 Datasets

The CASAS data[2] was collected in the Washington State University Smart D9epartment Testbed with multi-residents where each resident performs 15 unique activities [9]. The data is collected in 26 days in a smart home equipped with 37 ambient sensors

The ARAS data[3] [1] is collected in two different houses, denoted as House A and House B, in 30 days. In these environments, there are 20 sensors for two residents in each house.

6.2 Feature representations

There are several different ways to represent sensors' state, as follows. The easiest way is, one can treat each state of all sensors as a "word" in a vocabulary set which has the size of $\prod_i |s_i|$, where $|s_i|$ is the number of states of sensor i. Another representation method is to store the values of all sensors in a vector, as in [25]. However, one may argue that only a subset of sensors are triggered by human's activities at a time. Therefore, it is still able to use a vector to represent the observation, but an element i is set to 1 if and only if the i^{th} sensor changes its state, similar as [5]. Finally, we represent the sensors state as a one-hot vector where all elements are set to 0s except one element whose corresponding sensor

[2] http://ailab.eecs.wsu.edu/casas/

[3] http://www.cmpe.boun.edu.tr/aras/

Table 2 Evaluation results of six variants of HMMs using leave-one-out validation

		CASAS			ARAS House-A			ARAS House-B		
		R1	R2	All	R1	R2	All	R1	R2	All
pHMM	dis.	51.338	50.242	33.899	43.104	21.797	16.563	88.732	78.135	75.332
	vec1.	32.664	26.464	11.927	53.696	23.025	16.556	88.968	77.429	74.794
	vec2.	51.081	49.104	32.910	40.069	32.897	19.752	46.228	38.615	34.151
	vec3.	52.091	51.023	34.777	n/a	n/a	n/a	n/a	n/a	n/a
cHMM	dis.	62.661	63.887	46.321	43.724	23.969	17.110	88.823	78.049	75.306
	vec1.	32.733	26.618	12.780	43.716	37.336	17.036	89.051	77.473	74.821
	vec2.	63.986	64.461	47.131	39.747	32.917	20.080	45.091	38.119	34.099
	vec3.	64.858	64.289	48.183	n/a	n/a	n/a	n/a	n/a	n/a
gd-cHMM	dis.	70.760	69.144	59.330	43.606	24.024	17.039	88.752	78.101	75.305
	vec1.	34.023	27.811	13.860	43.247	37.347	16.964	88.987	77.521	74.822
	vec2.	70.554	68.670	58.907	39.588	32.812	19.885	49.140	46.496	40.247
	vec3.	72.852	70.794	61.839	n/a	n/a	n/a	n/a	n/a	n/a
fHMM	dis.	59.747	56.780	43.547	34.281	42.571	21.985	90.251	81.456	79.267
	vec1.	32.206	26.646	15.866	38.395	44.509	26.718	90.735	81.733	79.386
	vec2.	56.098	52.627	38.662	23.550	30.632	07.985	66.147	62.769	45.928
	vec3.	58.248	55.793	41.259	n/a	n/a	n/a	n/a	n/a	n/a
cd-fHMM	dis.	71.653	69.710	56.554	37.887	44.615	25.278	89.563	84.246	81.652
	vec1.	33.694	28.665	17.255	40.812	42.729	28.267	89.337	83.398	80.847
	vec2.	69.157	67.062	53.550	40.976	48.309	18.586	74.931	67.201	59.822
	vec3.	71.976	69.254	56.511	n/a	n/a	n/a	n/a	n/a	n/a
gd-HMM	dis.	77.368	78.267	69.114	38.017	45.398	25.487	89.449	84.318	81.717
	vec1.	34.346	29.882	18.403	40.692	43.333	28.250	89.119	83.456	80.802
	vec2.	75.617	75.188	66.542	41.236	48.056	18.520	79.546	71.039	67.382
	vec3.	77.706	77.540	69.127	n/a	n/a	n/a	n/a	n/a	n/a
md-HMM	dis.	57.939	60.567	47.314	45.369	52.850	31.857	90.227	86.035	83.173
	vec1.	34.415	28.055	17.095	46.600	50.588	33.331	89.918	85.023	82.285
	vec2.	56.542	58.887	45.210	48.292	62.438	31.475	82.241	78.610	73.664
	vec3.	59.307	61.388	47.889	n/a	n/a	n/a	n/a	n/a	n/a
MDM	dis.	77.791	**78.529**	69.335	**64.688**	**80.453**	**57.125**	**94.492**	**91.998**	**8**9.821
	vec1.	34.592	29.921	18.658	60.813	78.413	53.119	94.449	72.216	**90.109**
	vec2.	75.245	75.128	66.983	48.291	62.438	31.475	82.840	78.610	73.664
	vec3.	**78.459**	78.014	**69.865**	n/a	n/a	n/a	n/a	n/a	n/a

“R1”, “R2” indicate the average accuracy for each resident and “All” indicates the accuracy for all residents

has its value recorded along with the activities (no matter if the value is “ON” or “OFF”). This element is set to 1. This type of representation can only be obtained from the CASAS dataset. In Table 2 the notations “dis.”, “vec1.”, “vec2”, “vec3” indicate four different ways to represent the sensors’ states as discussed above respectively.

In the experiments we use leave-one-out cross validation for all datasets. In particular, the data of one day (one file) is employed for evaluation and the data of the other days are for training the models. We repeat the evaluation for every day and report the average accuracy.

6.3 Experimental results

6.3.1 Single dependency

In CASAS dataset, gd-HMM achieves much higher performance than the other variants with 69.127% accuracy. In comparison with other works which use the same evaluation method, in [18] the iterative CRF achieves 64.16% and in [6] pHMM achieves 61.78% accuracy. In [25] and [5], the authors report the accuracy of 60.60% and 75.77% respectively, but different from us they use threefold cross-validation. Also note that all these methods rely on the prior knowledge of data association while our HMMs do not. In ARAS House A and ARAS House B, cd-fHMM and gd-HMM achieve similar results which suggest that the current activities should be dependent on the combination of previous acitivities. The results in ARAS House A are small due to the complexity in its collected data, i.e. the number of available sensor states is ~ 9 times larger than CASAS and ~ 3 times larger than ARAS House B. For completeness, we also report the results in [1] with 61.5% and 76.2% for ARAS House A and ARAS House B respectively. Different from us, in that work the activities of each residents are grouped into 6 categories while we use all 27 activities. Overall, without data association the performance of parallel HMM drops dramatically. When coupling the activities using crossed dependencies, as in cHMM and cd-fHMM, we can observe improvement of performance. This means that activities of a resident indeed depend on the others'.

For feature representation, despite being simple discrete representation of sensors' states works very effectively in all three datasets. In practice the number of the states would grow exponentially with respect to the number of sensors. Fortunately, not all states are available, for example in CASAS, ARAS House A, and ARAS House B we have only 73, 655, and 200 states of sensors respectively. However, for a larger dataset with more residents, it would be more preferable to use vector representation for the sensors' states.

6.3.2 Mix dependency

From the results of six HMMs we find that modelling the interaction between residents and environment separately is not effective. Therefore we decide to construct an ensemble from fHMM, cdFHMM and gd-HMM, those consider the environment's state as a result of all residents' activities, as described in Section 4. Here α, β and γ are selected empirically. Two last rows in Table 2 show the results of ensemble model (md-HMM) and the mixed-dependency model (MDM) in comparison to the best accuracy from six HMMs extracted from Table 2. The results indicate that the ensemble model seems not very useful in CASAS data. This is because there are many misclassified activities from the parallel part which performs poorly in this case. In ARAS data, md-HMM performs very well and achieves better results than all variants. Combine with the performance of MDM, we can confirm our hypothesis that complexity of activities must be captured by multiple dependencies. Our MDM achieves impressive results in all three datasets, notably in ARAS data where it outperforms the others model with large margins. In particular, compare to the best HMMs from six variants MDM achieves higher accuracy of 0.738% in CASAS dataset, **28.858**% in ARAS house A, and **8.392**% in ARAS house B.

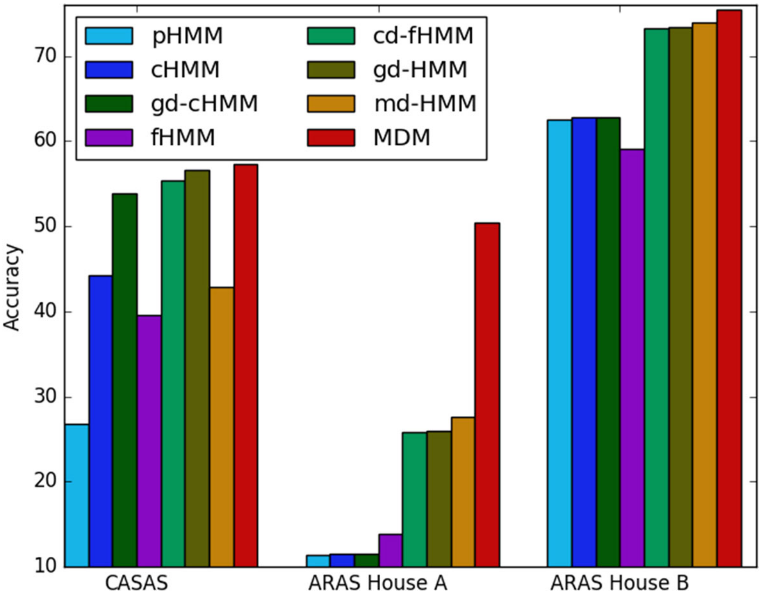


Fig. 1 Performance of all models on CASAS, ARAS House A and ARAS House B environments using model selection

6.3.3 Model selection

One concern over the selection of α, β and γ may be raised when applying MDM in practice. In order to show the effectiveness of our model we now evaluate MDM with a held-out test set. We apply grid-like search to select the α, β and γ using a separate validation set. In CASAS data we use 24 days for training, 1 day for validation and 1 day for testing. In ARAS House A and ARAS House B we partition the data into 10 days for training, 10 days for validation, and 10 days for testing. The results are shown in Fig. 1. As we can see, MDM performs better than other models. In CASAS, MDM is slightly better than the best result from other models (0.69%). In ARAS House B, MDM is 2.14% better than the best HMMs and 1.59% better than the ensembles. Especially in ARAS House A MDM achieves at least 22.80% higher than other models. However, we observe that MDM may get overfitting in the case of CASAS data if the training data size small. This makes sense because the activities in CASAS is much less complex than the activities in ARAS.

For completeness we also compare MDM with incremental decision trees [23]. Similar to them, for training we use the first 1, 7, 14 and 21 days and we test the models on days from 22 to 28. We use days 29 and 30 for model selection. The results in Fig. 2 show that MDM peforms better than IDT. With different number of days for training, in House A MDM has 0.86%, 2.52%, 4.18%, and 3.81% higher accuracy than IDT; and in House B MDM is 0.43%, 1.41%, 2.32% and 4.13% higher.

6.3.4 Compare with deep learning methods

Recently, deep learning has been employed for multi-resident activity recognition [22, 27]. In this experiment, we compare our MDM model with two notable deep learning models

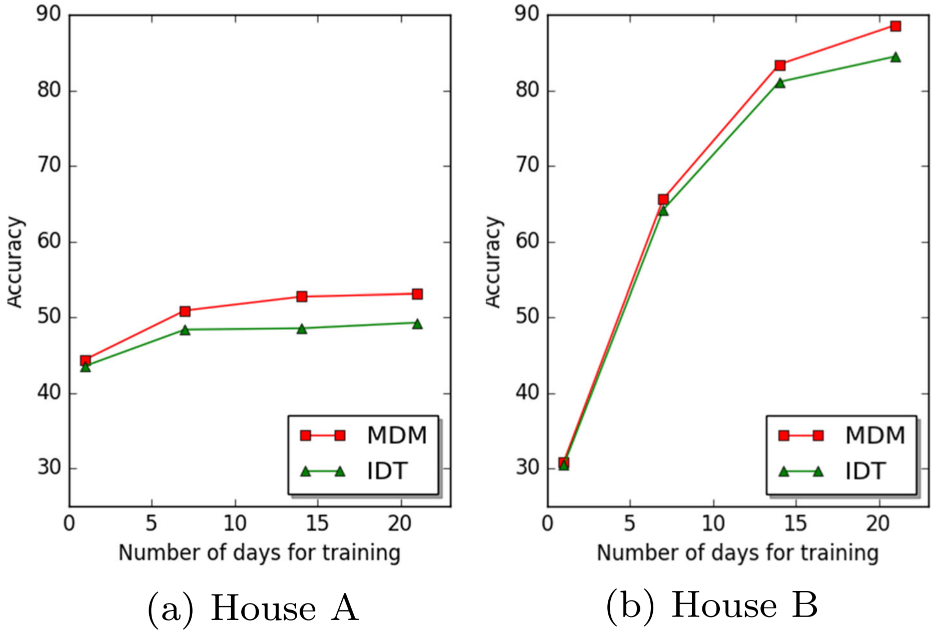


Fig. 2 MDM v.s IDT in ARAS House A & B

for sequence classification, namely Long Short-term Memory (LSTM) [17, 19] and Gated Recurrent Unit (GRU) [7]. In order to apply LSTM and GRU for the task in this paper we use multi ouputs to present activities of the residents. As we mentioned earlier, this work focuses on investigating the influence of multi-residents' activities on their future actions, not on performance improvement. HMM is used because it is simple and easy to explain the dependencies of the activities. As we can see in Table 3, its simplicity is also a disadvantage in comparison to complex deep learning models. It intrigues an idea for future work on incorporating mixed-dependency with deep recurrent neural networks.

7 Conclusions

This paper studies smart home environment with ambient settings, aiming to understand the insights of the behaviours of multiple residents. First, we break down the dynamics in such environment into activity dependencies and human-environment interaction. From that we construct six variants of HMMs for multi-resident activity modelling. We show that, good results can be achieved by using simple HMMs that capture the combined activities of all

Table 3 MDM versus LSTM and GRU

	ARAS A	ARAS B	CASAS
LSTM	64.25	74.19	77.49
GRU	65.16	75.28	77.91
MDM	50.43	75.03	57.27

residents. Second, the key contribution of the paper is our proposal of a mixed-dependency model to deal with the complexity of multiple residents' activities. The experimental results show that our model outperforms other dynamic Bayesian counterparts such as HMMs and CRFs which have been employed for activity recognition in ambient environment. Although the proposed MDM is less effective than deep learning approaches, it has shown a promising idea of incorporating mixed dependency with recurrent neural networks, which will be a future work.

References

1. Alemdar H, Ertan H, Incel OD, Ersoy C (2013) Aras human activity datasets in multiple homes with multiple residents. In: ICPCTH, PervasiveHealth '13, pp 232–235
2. Aminikhanghahi S, Cook D (2019) Enhancing activity recognition using cpd-based activity segmentation. Pervasive Mob Comput 53:75–89
3. Benmansour A, Bouchachia A, Feham M (2015) Multioccupant activity recognition in pervasive smart home environments. ACM Comput Surv 48(3):34:1–34:36. https://doi.org/10.1145/2835372
4. Brand M, Oliver N, Pentland A (1997) Coupled hidden markov models for complex action recognition. In: CVPR, CVPR '97, pp 994–
5. Chen R, Tong Y (2014) A two-stage method for solving multi-resident activity recognition in smart environments. Entropy 16(4):2184
6. Chiang YT, Hsu KC, Lu CH, Fu LC, Hsu JYJ (2010) Interaction models for multiple-resident activity recognition in a smart home. In: IEEE/RSJ international conference on IROS, pp 3753–3758
7. Chung J, Gülçehre Ç, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. ArXiv:1412.3555
8. Cook D (2012) Learning setting-generalized activity models for smart spaces. IEEE Intell Syst 27(undefined):32–38
9. Cook D, Crandall A, Singla G, Thomas B (2010) Detection of social interaction in smart spaces. Cybern Syst 41(2):90–104
10. Crandall AS, Cook D (2008) Resident and caregiver: handling multiple people in a smart care facility. In: Proceedings of AAAI fall symposium: AI in Eldercare: new solutions to old problems, AAAI Technical Report, vol FS-08-02. AAAI
11. Dahmen J, Cook D (2019) Synsys: a synthetic data generation system for healthcare applications. Sensors, 19
12. Dahmen J, Thomas BL, Cook D, Wang X (2017) Activity learning as a foundation for security monitoring in smart homes. Sensors, 17
13. Davis RIA, Lovell BC (2004) Comparing and evaluating HMM ensemble training algorithms using train and test and condition number criteria. Pattern Anal Appl 6(4):327–335
14. Feuz KD, Cook D (2017) Collegial activity learning between heterogeneous sensors. Knowl Inf Syst 53(2):337–364
15. Ghahramani Z, Jordan MI (1997) Factorial hidden Markov models. Mach Learn 29(2–3):245–273
16. Ghods A, Cook D (2019) Activity2vec: learning adl embeddings from sensor data with a sequence-to-sequence model. In: KDD Workshop on applied data science for healthcare
17. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
18. Hsu KC, Chiang YT, Lin GY, Lu CH, Hsu JYJ, Fu LC (2010) Strategies for inference mechanism of conditional random fields for multiple-resident activity recognition in a smart home. Springer, Berlin, pp 417–426
19. Mehr HD, Polat H, Cetin A (2016) Resident activity recognition in smart homes by using artificial neural networks. In: 2016 4th International Istanbul smart grid congress and fair (ICSG), pp 1–5. https://doi.org/10.1109/SGCF.2016.7492428
20. Minor B, Doppa JR, Cook D (2015) Data-driven activity prediction: algorithms, evaluation methodology, and applications. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '15. ACM, New York, pp 805–814. https://dx.doi.org/10.1145/2783258.2783408

21. Minor BD, Doppa JR, Cook D (2017) Learning activity predictors from sensor data: algorithms, evaluation, and applications. IEEE Trans Knowl Data Eng 29(12):2744–2757
22. Natani A, Sharma A, Peruma T, Sukhavasi S (2019) Deep learning for multi-resident activity recognition in ambient sensing smart homes. In: 2019 IEEE 8th Global conference on consumer electronics (GCCE), pp 340–341
23. Prossegger M, Bouchachia A (2014) Multi-resident activity recognition using incremental decision trees. In: Adaptive and intelligent systems - third international conference, ICAIS 2014, Bournemouth, UK, September 8-10, 2014. Proceedings, pp 182–191
24. Rabiner LR (1990) Readings in speech recognition. chap. A tutorial on hidden Markov models and selected applications in speech recognition. Morgan Kaufmann Publishers Inc., San Francisco, pp 267–296. http://dl.acm.org/citation.cfm?id=108235.108253
25. Singla G, Cook D, Schmitter-Edgecombe M (2010) Recognizing independent and joint activities among multiple residents in smart environments. J Ambient Intell Humaniz Comput 1(1):57–63. https://doi.org/10.1007/s12652-009-0007-1
26. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Proceedings of the 27th international conference on neural information processing systems - volume 2, NIPS'14. MIT Press, Cambridge, pp 3104–3112. http://dl.acm.org/citation.cfm?id=2969033.2969173
27. Tan T, Gochoo M, Huang S, Liu Y, Liu S, Huang Y (2018) Multi-resident activity recognition in a smart home using rgb activity image and dcnn. IEEE Sensors J 18(23):9718–9727
28. Tran SN, Zhang Q (2004) Towards multi-resident activity monitoring with smarter safer home platform. In: Chen F, García-Betances RI, Chen L, Cabrera-Umpiérrez MF, Nugent CD (eds) Smart assisted living, chap. 12. Springer International Publishing, Springer, pp 249–267
29. Tunca C, Alemdar H, Ertan H, Incel OD, Ersoy C (2014) Multimodal wireless sensor network-based ambient assisted living in real homes with multiple residents. Sensors 14:9692–9719
30. Wang L, Gu T, Tao X, Chen H, Lu J (2011) Recognizing multi-user activities using wearable sensors in a smart home. Pervasive Mob Comput 7(3):287–298. https://doi.org/10.1016/j.pmcj.2010.11.008
31. Wang A, Chen G, Shang C, Zhang M, Liu L (2016) Human activity recognition in a smart home environment with stacked denoising autoencoders. In: WAIM Workshops

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Dr. Son N. Tran is a Lecturer (Assistant Professor) at the University of Tasmania, Australia. Before that, Son was a post-doctoral research fellow at the Commonwealth Scientific and Industrial Research Organisation, Australia. He holds a Ph.D. in Computer Science (2016) from City, University of London, an EU Erasmus Mundus joint MSc in Networks and e-Business Computing, University of Reading, and a BSc in Electronic Engineering from Hanoi University of Science and Technology. Son is the recipient of the Erasmus Mundus Outstanding Academic Achievement award (2010) and the AES producible prize (2014). He published more than 30 peer-reviewed papers and has been a reviewer for top-tier conferences and journals (AAAI, IJCAI, ACL, IEEE TNNLS, NN, IEEE CYB). His research interests are AI and applications of AI and IoT for health care.

Tung-Son Ngo is a lecturer at FPT University, Vietnam. He has two years experience working as a software engineer and a R&D engineer at Panasonic laboratory in Vietnam. Son Ngo received a MSc in Computer Science University of Lorraine, Metz , France.

Dr. Qing Zhang received the BSc degree from Tsinghua University, Beijing, China, and the PhD degree in computer science from University of New South Wales, Sydney, Australia. He is now a senior research scientist at the Australian e-Health Research Centre, CSIRO and leads the smart home and internet of things projects. He has served as the committee member of IEEE QLD section. His research interests span a variety of topics in sensor data fusion and mining, human identification and activity recognition.

Dr. Mohanraj Karunanithi obtained his doctorate in Biomedical Engineering at the University of New South Wales. He has worked in cardiac research for 10 years on ventricular function. He is a Group Leader at the AEHRC, CSIRO coordinating and directing the research of projects, scientists, and students in the area of Integrated mobile and telehealth health applications in the management of chronic diseases, aged care and independently living. He is a senior member of the largest biomedical engineering society, IEEE Engineering Medicine and Biology Society (EMBS). He is also the founding and current chair of the IEEE EMBS chapter in Queensland, Australia.