Mixed-Dependency Models for Multi-resident Activity Recognition in Smart-Homes

SON N. TRAN, The Australian E-Health Research Centre, CSIRO
QING ZHANG, The Australian E-Health Research Centre, CSIRO
MOHAN KARUNANITHI, The Australian E-Health Research Centre, CSIRO

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 comprehensive 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 which leads to the 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.

CCS Concepts: •**Human-centered computing** → **Ubiquitous and mobile computing;** *Ubiquitous and mobile computing theory, concepts and paradigms; Ambient intelligence;*

Additional Key Words and Phrases: Multi-resident Activity Mornitoring, Smart Homes, Temporal Modelling

ACM Reference Format:

Son N. Tran, Qing Zhang and Mohan Karunanithi. 2017. Hidden Markov Models for Multi-resident Activity Recognition in Smart Homes. *ACM Trans. Appl. Percept.* 2, 3, Article 1 (May 2016), 16 pages.
DOI: 0000001.0000001

1. INTRODUCTION

In intelligent environments such as smart homes activity recognition plays an important role, especially when apply to health monitoring and assistance [Chernbumroong et al. 2013; Mocanu et al. 2011; Das and Cook 2004]. Much effort has been spent on modelling the activities of residents in order to facilitate reasoning of their behaviour. The success of such models would result in reducing cost of traditional health care, a smarter and safer home for eldercare, and better assistance for patients. Classifying human activities has been studied intensively within computer vision domain [Poppe 2010]. This, however, may raise an issue on privacy of residents due to the need of unwelcome devices, i.e. cameras. Alternatively, many other approaches rely on wearable sensors [Plötz et al. 2011; Liu et al. 2016; Liu et al. 2015], which seems less intrusive but are very inconvenient and uncomfortable for users. Recent attention is aiming at intelligent environments where residents can live their own way,

Author's address: S. N. Tran, Q. Zhang and M. Karunanithi, The Australian E-Health Research Centre, Level 5 UQ Health Sciences Building Royal Brisbane and Women's Hospital Herston, Queensland 4029 Australia, email: {son.tran,qing.zhang,mohan.karunanithi}@csiro.au.

 1544-3558/2016/05-ART1 $15.00
DOI: 0000001.0000001

without being disturbed by the presence of the devices. This is an important research topic that would shape the future of smart homes. With the advance in pervasive sensing technologies one can install a set of non-intrusive sensors in the environment with respect to residents' privacy [Wilson and Atkeson 2005; Singla et al. 2010]. However, in contrast with the development of ambient hardware, the reality of intelligent algorithms for such modern smart homes is still challenging.

Activity recognition in ambient environment has been studied for years, most of that focuses on single resident, aiming to support independent living [van Kasteren et al. 2008]. However, in practice this is not always the case since modern smart environments should be able to support multiple occupants. As a result, there is a growing desire for a model that is capable of capturing the complexity nature of both independent and joint activities. This is a challenging task because different from the case of single resident where the sensors' states reflect directly the activity of a specific person, in multi-resident case that information, as known as *data association*, however is not commonly known. In recent work, temporal approaches have been widely employed to model activities in smart homes [Singla et al. 2010; Cook 2012; Chen and Tong 2014; Alemdar et al. 2013; Chiang et al. 2010; Wang et al. 2011]. However, there still lacks a comprehensive study to fully understand the relations of residents' behaviours and how they are reflected through the sensors' states. In this paper, we show that such study is helpful not only in understanding the complexity of multi-resident activity modelling in ambient environment, but also in developing a novel model to improve prediction performance.

The first contribution of this paper is an investigation into the behaviour dependencies in smart home environments. For single resident, daily-living patterns are easy to model and reason through the interactions between the resident and the environment, as being recorded by the sensors' states. For multiple resident the task becomes more challenging since it is difficult to map the sensors' states to the activities of each person. Moreover, there exist some forms of correlation between the activities of residents. For example, when one is talking on the phone the others should be doing something else. Also, cooperative activities such as moving a table, playing checker, etc. should be taken into account thoroughly. We, therefore, 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.

The second contribution 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 multiple, interactive activities. We then propose an ensemble of HMMs, which we call as 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. We find that the role of each type of dependency varies in different environments, depending on the living styles of the 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 conduct the experiments on three smart home environments from CASAS [Cook et al. 2010; Hsu et al. 2010] and ARAS [Alemdar et al. 2013] 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 public 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 the 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 while MDM achieves further improvement with a magnitude.

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 generalization 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. Although comparing to cameras and wearable sensors several types of ambient sensors provide lower quality data, we can compensate this by use more sensors with various types to enrich the information and employing probabilistic models to deal with noise and uncertainty.

2.1 Notations

Let us denote $a^{m,t}$ and o^t as the activity of resident m and the 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 the 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 the sequence of activities performed by all residents from time t_1 to t_2.

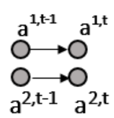
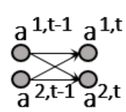
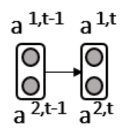
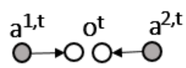
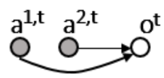
			
	pHMM	cHMM	gd-cHMM
	fHMM	cd-fHMM	gd-HMM

Table I. : 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

2.2 Activity Dependencies

In the case of single resident, a sequence of activities can be viewed as a Markov chain where an activity at time t depends on the previous activity $t-1$. For multiple residents there are three types of dependencies as being illustrated in the first row of Table I. 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 I. 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 activate them. Instead, the 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 previous section. Here, we can either use Conditional Random Fields (CRFs) or Hidden Markov Models. We find that in previous works on the same problem as in this paper, CRFs do not show apparent advantage over HMMs [Cook 2012; Wang et al. 2011]. Therefore, we choose HMMs for efficient learning and reasoning. As the result, this leads to the construction of six different HMMs.

3.1 Hidden Markov Models

A HMM [Rabiner 1990] 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 observation, 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 I. 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 [Chiang et al. 2010]. 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 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 [Chiang et al. 2010]. 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.

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

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 [Ghahramani and Jordan 1997]. 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

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

except that there is only a single observation. We take into account that in pHMM and cHMM the sensors depend on each individuals 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:

$$p = p(o^1|\mathbf{a}^1)\prod_m p(a^{m,1})\prod_{t=2}^{T}\left[p(o^t|\mathbf{a}^t)\prod_m p(a^{m,t}|a^{m,t-1})\right] \quad (5)$$

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 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 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:

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

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:

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

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 $\mu_t = \mathbf{max}_{\mathbf{a}^{1:t-1}} p(\mathbf{a}^t = j, \mathbf{a}^{1:t-1}, o^{1:t})$ we have:

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

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 Eq. 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). The combined log-probability of this model is:

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

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.

Inference in MDM is efficient as in other HMMs and md-HMM with different $\mu_t(j)$ as:

$$\begin{aligned}\mu_t(j) = \log(p(o^t|\mathbf{a}^t = j)) + \mathbf{max}_{j'}[&\alpha \log p(\mathbf{a}^t|\mathbf{a}^{t-1})\\ &+ \sum_m \beta \log p(a^{m,t}|\mathbf{a}^{t-1})\\ &+ \gamma \log p(a^{m,t}|a^{m,t-1})\\ &+ \mu_{t-1}(j')]\end{aligned} \tag{12}$$

5. RELATED WORK

Hidden Markov Models [Rabiner 1990] is a popular statistical model for sequential data. It is characterized by the dependency of an observation variable on a hidden variable at each time step, and the dependency of the hidden variable itself on its previous state. HMMs can be employed for activity recognition easily. In particular, one can define the observation as the sensors state, i.e. video frame, wearable or/and ambient sensors' values, and the hidden variable as the activity [Kim et al. 2010].

In multi-resident smart homes, HMMs have been studied intensively, as being showed in previous studies [Alemdar et al. 2013; Chen and Tong 2014; Singla et al. 2010; Cook 2012]. 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 [Chen and Tong 2014].

Another method to model the activities of multiple residents is to create multiple HMMs, one for each resident [Chiang et al. 2010]. Such model, as known as parallel HMM, has been evaluated in the case that data association is provided. This means that the observation has been separated for each resident and only represents the sensors which are associating to that resident. The disadvantage of this model is the hidden variables of all HMMs are independent from each others. In multi-resident environments, however, there always exist correlation and interaction between the residents. This issue is addressed by adding the crossed dependencies to the hidden variables in all HMMs. By coupling such HMMs one can assume that the activity of a resident is dependent not only on his previous activity but also on the previous activities of other residents. There was a proposal of coupled HMM and factorial HMM in computer vision domain [Brand et al. 1997], but only cHMM was employed for sensor data [Chiang et al. 2010].

Combining HMMs of the same type was studied in [Davis and Lovell 2004]. Different from that, in this paper we ensemble HMMs of various types for activity recognition.

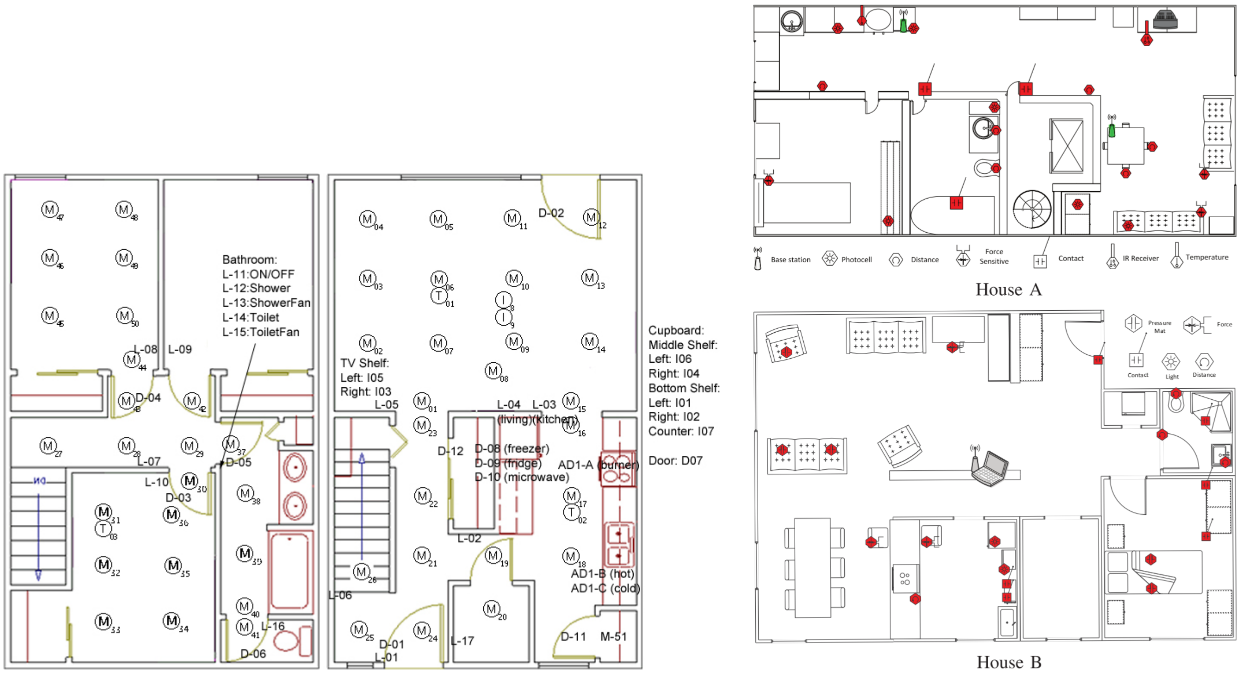
Besides HMMs, CRFs [Crandall and Cook 2008; Hsu et al. 2010] and incremental decision trees (IDT) [Prossegger and Bouchachia 2014] also have been used for multi-resident activity recognition . We also discuss their performances in the experiments.

6. DATASETS

6.1 CASAS

The CASAS data[2] was collected in the WSU smart department Testbed with multi-residents where each resident performing 15 unique activities [Cook et al. 2010]. All the activities are described in Table II. The data is collected in 26 days in a smart home equipped with 37 ambient sensors, the layout can be seen in Figure 1a. Data in CASAS is presented in "Date Time Sensor_ID Value Resident_ID Activity" format. For example, "2008-11-10 14:28:17.986759 M22 ON 2 2" shows that resident 2 is hanging up clothes at 14:28:17.986759 on 2008-11-10 when motion sensor M22 is triggered. Similarly, "2008-11-10

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



(a) CASAS dataset

(b) ARAS dataset

14:38:47.974299 M13 OFF 2 8 1 9" means at 14:38:47.974299 on 2008-11-10 when motion sensor is off resident 1 is setting dining room table for dinner while resident 2 is setting out ingredients for dinner in the kitchen.

6.2 ARAS

The ARAS data[3] [Alemdar et al. 2013] 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, see the layout in Figure 1b. Each resident is ask to perform 27 different activities, as in Table III. The format of ARAS data is presented as: "Sensor_1 Sensor_2 .... Sensor_20 R1_Activity R2_Activity". This can be seen as a vector of sensors' values and activities of residents. For example, "0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 11 16" indicates that resident 1 is watching TV and resident 2 is using internet when the force sensor in the couch (sensor 4) and distance sensor in the chair (sensor 6) are triggered.

6.3 Data Analysis

In order to understand the behaviour of residents we visualise their activities in three environments, as being shown in Figure 2 . The x and y axes indicate the activities of resident 1 and resident 2 respectively. The hotter color of the cell indicates that the activities of two residents occur more often (see the heat map bar in each figure).

In CASAS data, there exist many cases that only individual's activities are labelled, i.e. only activity of one resident is known at a time. Here we use activity ID = 0 to present unknown activity of the other resident. As we can see in Figure 2a, there is only 5 cases where two residents perform different

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

ID	Activity	Description
1	Filling medication dispenser.	Fill medication dispenser in the kitchen using items obtained from the cabinet. Return items to the cabinet when done.
2	Hanging up clothes.	Hang up clothes in the hallway closet. The clothes are laid out on the couch in the living room.
3	Moving furniture.	Move the couch and coffee table to the other side of the living room. Request help from other person.
4	Reading magazine .1	Sit on the couch and read a magazine.
5	Watering plants.	Water plants located around the apartment. Use the watering can located in the hallway closet. Return the watering can to the closet when finished.
6	Sweeping floor.	Sweep the kitchen floor using the broom and dust pan located in the kitchen closet. Return the tools to the closet when finished.
7	Playing checker.	Play a game of checkers for a maximum of five minutes.
8	Preparing dinner.	Set out ingredients for dinner in the kitchen.
9	Setting table.	Set dining room table for dinner.
10	Reading magazine 2.	Read a magazine on the living room couch.
11	Paying bill.	Simulate paying electric bill.
12	Gathering food.	Gather food for a picnic from the kitchen cupboard and pack them in a picnic basket.
13	Retrieving dishes.	Retrieve dishes from a kitchen cabinet.
14	Packing supplies.	Pack supplies in the picnic basket.
15	Packing food.	Pack food in the picnic basket and bring the basket to the front door of the apartment.

Table II. : Activities of each resident in CASAS dataset.

ID	Activity	ID	Activity	ID	Activity
0	Other	1	Going Out	2	Preparing Breakfast
3	Having Breakfast	4	Preparing Lunch	5	Having Lunch
6	Preparing Dinner	7	Having Dinner	8	Washing Dishes
9	Having Snack	10	Sleeping	11	Watching TV
12	Studying	13	Having Shower	14	Toileting
15	Napping	16	Using Internet	17	Reading Book
18	Laundry	19	Shaving	20	Brushing Teeth
21	Talking on the Phone	22	Listening to Music	23	Cleaning
24	Having Conversation	25	Having Guest	26	Changing Clothes

Table III. : Activities of each resident in ARAS dataset.

activities at the same time. Notably, there is only one case where they are doing the same activity which is cooperative task where a resident retrieves dishes from a kitchen cabinet and resident 2 requests help from resident 1 to identify cabinet in which the dishes are located.

In ARAS house A and ARAS house B the distributions of activities are much less biased than in the CASAS with many different activities have been performed and their occurrences are almost similar. Also, in these environments more cooperative tasks can be seen too, such as “using internet”, “sleeping”, “going out”, “having dinner”, “having breakfast”, etc.

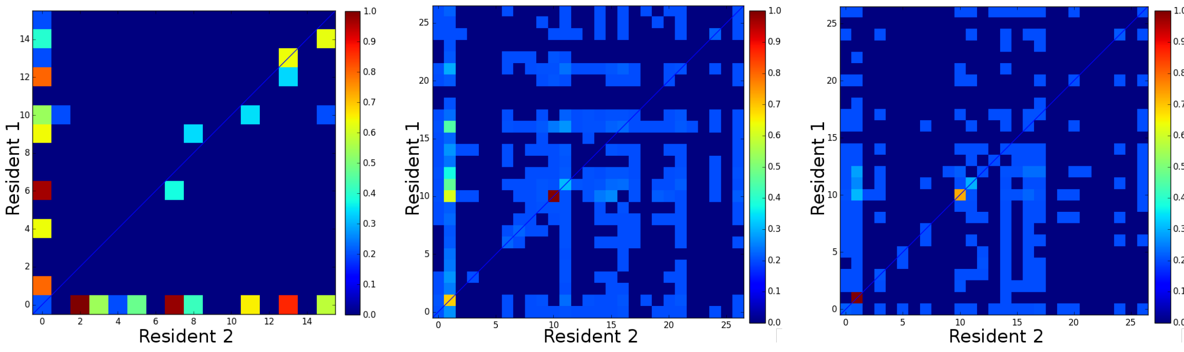


Fig. 2: Distributions of residents' activities in CASAS, ARAS House A and ARAS House B

7. EXPERIMENTS

7.1 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. By doing this, we have defined the sensors state as a set of discrete values and therefore being suitable for HMMs with discrete observation. Another representation method is to store the values of all sensors in a vector. Each sensor is represented as an element in the vector whose value will be updated according to the state of that sensor. This type of representation was used in [Singla et al. 2010]. 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 [Chen and Tong 2014]. Finally, we represent the sensors state as a one-hot vector where all elements are set to 0s except one element whose corresponding sensor 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 IV and Table V, the notations "dis.", "vec1.", "vec2", "vec3" indicate four different ways to represent the sensors' states as discussed above. In particular, "dis" is the discrete representation of sensors; "vec1." is a vector representation where each element is the value of a sensor; "vec2." is a binary vector representation where an element is set to 1 if and only if the corresponding sensor changes its state; "vec3." is a one-hot vector representation where an element is set to 1 if and only if the corresponding sensor have its state recorded when residents perform activities.

7.2 Experimental Results

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.

7.2.1 *Single dependency.* In CASAS dataset, gd-HMM achieves much higher performance than the other variants with 69.127% accuracy. In comparison with other work which use the same evaluation method, in [Hsu et al. 2010] the iterative CRF achieves 64.16% and in [Chiang et al. 2010] pHMM achieves 61.78% accuracy. In [Singla et al. 2010] and [Chen and Tong 2014], the authors report the accuracy of 60.60% and 75.77% respectively, but different from us they use threefold cross-validation. Also

Model	Data	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

Table IV. : Evaluation results of six variants of HMMs using leave-one-out validation. "R1", "R2" indicate the average accuracy for each resident and "All" indicates the accuracy for all residents.

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 is 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 [Alemdar et al. 2013] 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 the 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.

One may notice that in CASAS, gd-HMM is better than cd-fHMM while in ARAS House A and ARAS House B these two models perform similarly. This is because we denote the unknown activity in CASAS as "0" as shown in 6.3 which makes the cross dependency $p(a^{m,t} = 0|\mathbf{a}^{t-1} = j)$ ambiguous. Normally, this issue is nontrivial if the data is unbiased. However, in this dataset as we can see in Figure 2a that the unknown activity overwhelms other types of activities which causes the problem. For gd-HMM this is less severe since $p(a^t = j'|\mathbf{a}^{t-1} = j)$ can rule out the ambiguity of a resident's unknonw activity by focusing on the real activity of the other. In ARAS House A and ARAS House B, such problem seems not happen thanks to the various types of activities have been labelled (see Figure 2b and Figure 2c).

7.2.2 *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. Table V shows 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 IV. 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, $\mathbf{28.858}\%$ in ARAS house A, and $\mathbf{8.392}\%$ in ARAS house B.

HMM (best)	dis.	77.368	78.267	69.114	43.724	45.398	25.487	90.251	84.318	81.717
	vec1.	34.346	29.882	18.403	53.696	44.509	28.267	90.735	83.456	80.847
	vec2.	75.617	75.188	66.542	48.309	62.438	20.080	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**	89.821
	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

Table V. : Evaluation results of md-HMM and MDM using leave-one-out validation. "R1", "R2" indicate the average accuracy for each resident and "All" indicates the accuracy for all residents. HMM (best) is the best results extracted from all six variants of HMMs.

7.2.3 *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 Figure 3. 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.

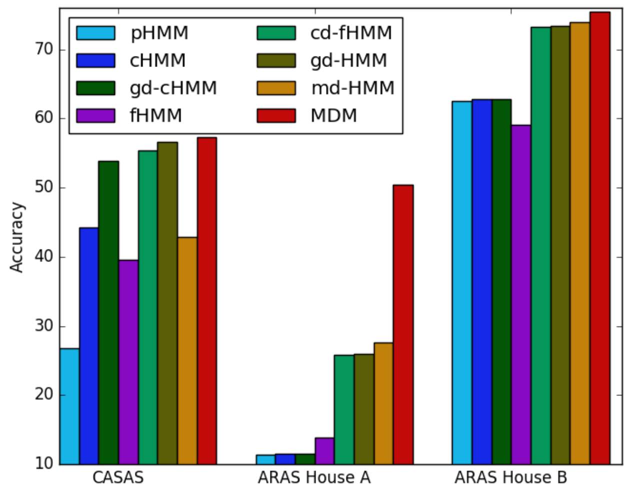
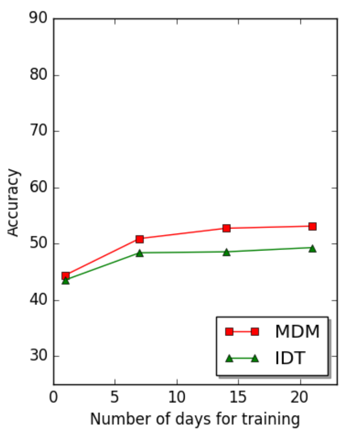
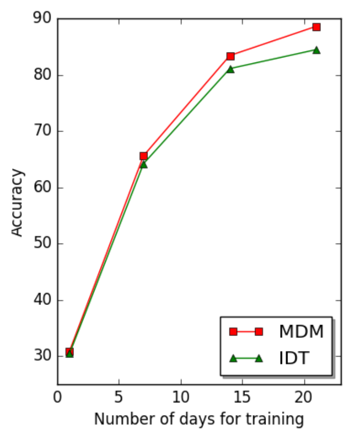


Fig. 3: Performance of all models on CASAS, ARAS House A and ARAS House B environments using model selection

Finally, for completeness we also compare MDM with incremental decision trees [Prossegger and Bouchachia 2014]. 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 Figure 4 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.



(a) House A



(b) House B

Fig. 4: MDM v.s IDT in ARAS House A & B

8. 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 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 all other models which have been employed for activity recognition in ambient environment.

The idea of mixed-dependency in this paper should be well fitted with CRFs where different dependencies can be seen as different type of features. We can also generalize this work to high-order version of HMMs/CRFs.

REFERENCES

Hande Alemdar, Halil Ertan, Ozlem Durmaz Incel, and Cem Ersoy. 2013. ARAS Human Activity Datasets in Multiple Homes with Multiple Residents. In *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth '13)*. ICST, ICST, Brussels, Belgium, Belgium, 232–235. DOI:http://dx.doi.org/10.4108/icst.pervasivehealth.2013.252120

M. Brand, N. Oliver, and A. Pentland. 1997. Coupled Hidden Markov Models for Complex Action Recognition. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*. IEEE Computer Society, Washington, DC, USA, 994–.

Rong Chen and Yu Tong. 2014. A Two-stage Method for Solving Multi-resident Activity Recognition in Smart Environments. *Entropy* 16, 4 (2014), 2184. http://www.mdpi.com/1099-4300/16/4/2184

Saisakul Chernbumroong, Shuang Cang, Anthony Atkins, and Hongnian Yu. 2013. Elderly activities recognition and classification for applications in assisted living. *Expert Systems with Applications* 40, 5 (2013), 1662 – 1674. DOI:http://dx.doi.org/10.1016/j.eswa.2012.09.004

Yi-Ting Chiang, K. C. Hsu, C. H. Lu, Li-Chen Fu, and Jane Yung-Jen Hsu. 2010. Interaction models for multiple-resident activity recognition in a smart home. In *IEEE/RSJ International Conference on IROS*. 3753–3758. DOI:http://dx.doi.org/10.1109/IROS.2010.5650340

Diane J. Cook. 2012. Learning Setting-Generalized Activity Models for Smart Spaces. *IEEE Intelligent Systems* 27, undefined (2012), 32–38. DOI:http://dx.doi.org/doi.ieeecomputersociety.org/10.1109/MIS.2010.112

Diane J. Cook, Aaron Crandall, Geetika Singla, and Brian Thomas. 2010. DETECTION OF SOCIAL INTERACTION IN SMART SPACES. *Cybern. Syst.* 41, 2 (Feb. 2010), 90–104. DOI:http://dx.doi.org/10.1080/01969720903584183

Aaron S. Crandall and Diane J. Cook. 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. http://www.aaai.org/Library/Symposia/Fall/fs08-02.php

Sajal K. Das and Diane J. Cook. 2004. Health Monitoring in an Agent-Based Smart Home. In *In Proceedings of the International Conference on Smart Homes and Health Telematics (ICOST*. IOS Press, 3–14.

Richard I. A. Davis and Brian C. Lovell. 2004. Comparing and evaluating HMM ensemble training algorithms using train and test and condition number criteria. *Pattern Anal. Appl.* 6, 4 (2004), 327–335. http://www.springerlink.com/index/10.1007/s10044-003-0198-6

Zoubin Ghahramani and Michael I. Jordan. 1997. Factorial Hidden Markov Models. *Mach. Learn.* 29, 2-3 (Nov. 1997), 245–273. DOI:http://dx.doi.org/10.1023/A:1007425814087

Kuo-Chung Hsu, Yi-Ting Chiang, Gu-Yang Lin, Ching-Hu Lu, Jane Yung-Jen Hsu, and Li-Chen Fu. 2010. *Strategies for Inference Mechanism of Conditional Random Fields for Multiple-Resident Activity Recognition in a Smart Home*. Springer Berlin Heidelberg, Berlin, Heidelberg, 417–426. DOI:http://dx.doi.org/10.1007/978-3-642-13022-9_42

Eunju Kim, Sumi Helal, and Diane Cook. 2010. Human Activity Recognition and Pattern Discovery. *IEEE Pervasive Computing* 9, 1 (Jan. 2010), 48–53. DOI:http://dx.doi.org/10.1109/MPRV.2010.7

Li Liu, Li Cheng, Ye Liu, Yongpo Jia, and David Rosenblum. 2016. Recognizing Complex Activities by a Probabilistic Interval-Based Model.

Ye Liu, Liqiang Nie, Lei Han, Luming Zhang, and David S. Rosenblum. 2015. Action2Activity: Recognizing Complex Activities from Sensor Data. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15)*. AAAI Press, 1617–1623. http://dl.acm.org/citation.cfm?id=2832415.2832474

Stefan Mocanu, Irina Mocanu, Silvia Anton, and Calin Munteanu. 2011. AmIHomCare: A Complex Ambient Intelligent System for Home Medical Assistance. In *Proceedings of the 10th WSEAS International Conference on Applied Computer and Applied*

Computational Science (ACACOS'11). WSEAS, Stevens Point, Wisconsin, USA, 181–186. http://dl.acm.org/citation.cfm?id=1965610.1965641

Thomas Plötz, Nils Y. Hammerla, and Patrick Olivier. 2011. Feature Learning for Activity Recognition in Ubiquitous Computing. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two (IJCAI'11)*. AAAI Press, 1729–1734.

Ronald Poppe. 2010. A Survey on Vision-based Human Action Recognition. *Image Vision Comput.* 28, 6 (June 2010), 976–990.

Markus Prossegger and Abdelhamid Bouchachia. 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*. 182–191.

Lawrence R. Rabiner. 1990. Readings in Speech Recognition. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, Chapter A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, 267–296. http://dl.acm.org/citation.cfm?id=108235.108253

Geetika Singla, Diane J. Cook, and Maureen Schmitter-Edgecombe. 2010. Recognizing independent and joint activities among multiple residents in smart environments. *Journal of Ambient Intelligence and Humanized Computing* 1, 1 (2010), 57–63. DOI:http://dx.doi.org/10.1007/s12652-009-0007-1

Tim van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. 2008. Accurate Activity Recognition in a Home Setting. In *Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp '08)*. ACM, New York, NY, USA, 1–9. DOI:http://dx.doi.org/10.1145/1409635.1409637

Liang Wang, Tao Gu, Xianping Tao, Hanhua Chen, and Jian Lu. 2011. Recognizing Multi-user Activities Using Wearable Sensors in a Smart Home. *Pervasive Mob. Comput.* 7, 3 (June 2011), 287–298. DOI:http://dx.doi.org/10.1016/j.pmcj.2010.11.008

D. H. Wilson and C. Atkeson. 2005. Simultaneous Tracking and Activity Recognition (STAR) Using Many Anonymous, Binary Sensors. In *Proceedings of the Third International Conference on Pervasive Computing (PERVASIVE'05)*. Springer-Verlag, Berlin, Heidelberg, 62–79. DOI:http://dx.doi.org/10.1007/11428572_5