Mapping of Activity Recognition as a Distributed Inference Problem in Sensor Network

Md. Kamrul Hasan[1], Husne Ara Rubaiyeat[2], Young-Koo Lee[1], Sungyoung Lee[1]
[1]Ubiquitous Computing Lab, Department of Computer Engineering,
[2]Bio Imaging Lab, Department of Biomedical Engineering,
Kyung Hee University, 1 Socheon, Giheung, Yongin, Gyeonggi, 446-701, South Korea
kamrul@oslab.khu.ac.k, rubaiyeat@yahoo.com, yklee@khu.ac.kr, sylee@oslab.khu.ac.kr

Abstract - *This paper presents a distributed model for detecting Activities of Daily Living (ADLs) in a home setting. We consider an environment where household devices and utensils are augmented with simple sensors. The variation of object usage and shortage of training data makes it infeasible to construct high level models for ADLs in this environment. However, ADLs produce a series of sensor activations and the sensor activation sequences can help identifying the activity being performed. Hence, we try to correlate the sensor activation sequences with the activities. We provide a simple distributed linear time inference algorithm for this. The algorithm scales well with the size of the sensor network and number of activities to be detected.*

Keywords: Distributed Activity Recognition, Activity of Daily Living, Hidden Markov Model

1 Introduction

In ubiquitous healthcare environment, it is essential to recognize user's Activities of Daily Living (ADLs) like, preparing meal, bathing, etc. [1]. Recognizing and monitoring the daily activities can provide proactive healthcare at home [2]. Monitoring can reduce the number of low risk cases that come to hospitals-thereby reducing the financial burden on health systems and allowing more focus on patients at higher risk.

ADLs involve the household devices and utensils. A user manipulates and uses the devices and utensils in a certain fashion. Still there is so much variation in doing an activity that it is very difficult to learn all the models due to the shortage of enough training data [3]. In addition to this, there can be noise in the data as well, as user's movements are not always purposeful, and even if purposeful, the movement can be for performing a different activity. Figure 1 shows a deviation of the user from the default course of movement.

So, we move away from using a high level model defining the networks for ADLs involving all possible combination of daily object usage. We rather focus on using the temporal relationships among the sensors. As for example, a 'preparing meal' activity is indicated by 'going to the kitchen', 'turning on the gas burner', and so on. Naturally, those sensors attached to the cooking utensils will be activated while the user is preparing a meal. Some other sensors can also be activated during that period as the user deviates from meal preparation. Finding the most relevant sensors and ignoring other sensor events, can contribute to the detection of the user activity.

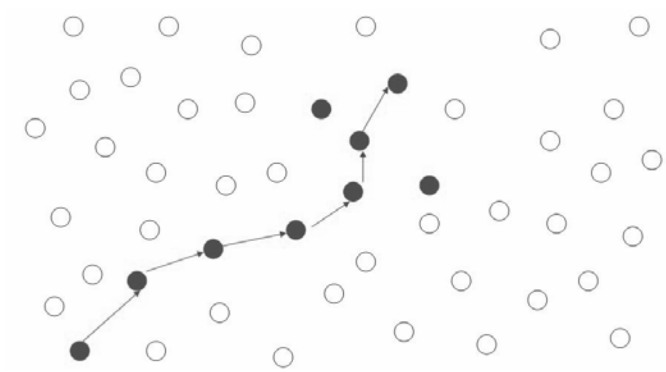

Figure 1. Example of sensor activations as user deviates from his default course of movement. Circles represent sensor and red circles represent activated sensors. Red circles connected through arrow are the default route.

Current activity recognition systems are centralized in approach [1], [2], [4]. Sensor events are gathered to a central server and a monolithic reasoning engine infers the activities. This incurs communication and computation overhead on the central server. Space complexity of any network grows exponentially with the number of discrete values of the random variables. The time complexity is quadratic to exponential in nature with the additional step of making features which may grow polynomial [5]. It is possible that the space and time complexity become intractable for a complex network structure [1], [6]. As for example, if not distributed, a Hidden Markov Model (HMM) having N states will need N^N state transition probabilities, 2^N output probabilities (assuming all the outputs are binary) and N^2L time complexity to derive the probability of an output sequence of length L [7]. So, to restrict the time and space requirements and distribute the computation, it is needed to devise distributed algorithm.

We take a distributed probabilistic approach for activity recognition from activation sequence of simple sensors deployed in the environment. We show that activity recognition by using temporal relationships among sensor can be implemented in a distributed fashion, providing each sensor a role. The only thing each sensor needs to store is a set of probabilities. Each sensor stores its own prior probability and conditional probabilities with respect to other sensors, for each activity. A sensor stores conditional probability only for those sensors that can be activated before it. The prior and conditional probabilities stored are local to each sensor and using them, a sensor can calculate the likelihood for an activity.

A sensor broadcasts its state as it is activated. Other sensors receive the broadcasts and make a list of activated sensors. Sensors make a new sequence after a sampling time period. A new entry is given for a sensor which broadcasted its activation in the last sampling and has not broadcasted its deactivation yet. So, each sensor maintains a sequence of sensors activated in last few sampling times. With the sensor sequence we construct a virtual Hidden Markov Model for each activity to be recognized. We map each sensor to a state of the HMM; doing so, distributes the calculation for each state of HMM to a sensor itself. We consider HMM construct because it can be visualized graphically and it naturally captures the time sequence of the states i.e., the sensor activations and discards the necessity for making temporal features. To the best of our knowledge, we are the first to provide a distributed inference mechanism for activity recognition.

2 Related Works

Distributed activity recognition is quite a new concept. The first approach of this kind is found in [5]. They use the concept of hierarchical feature extraction from tracking data, such as, 'the cluster head detects in which direction the user is moving'. Their core inference mechanism is Naïve Bayes with temporal features as input. Compared to their approach, our method focuses on a distributed inference mechanism. We focus on in-network processing of feature values and the final outcome, namely activity. We are the first to provide a distributed reasoning mechanism for activity recognition.

Activity Recognition for a setup where sensors are deployed in the environment was addressed in [1], [8]. They used 77 on/off state sensors in a house to capture the daily activities. They got very low recognition accuracy (27% maximum) on the data collected. Even though they tried to incorporate temporal relationships, the accuracy decreased while increasing the number of features. In our experiment, we use the same data and show that temporal relationships among features actually increase the accuracy up to a certain limit. We also show that in the same setup it is possible to use a distributed reasoning for detecting the ADLs.

Mobility based activity detection is a simplified version of activity recognition problem, where only user's motion is considered. Several approaches have been found in mobility based activity detection, such as Hierarchical HMM [4], Bayes Filter [2], Dynamic Bayesian Network [9], Naïve Bayes [5] etc. Except [5] all are centralized in approach and concentrate on special techniques to use the specific domain knowledge. Our work provides a general distributed model and shows an example how to use the approach for activity detection in one of the setups. All those works use temporal features created explicitly or achieved through placing several time slices of the same network sequentially. Those approaches need huge computation and may become intractable, if not carefully designed [1], [6]. On the other hand, our approach relates features in time sequence naturally through the model proposed.

We are motivated by the work [10] that motivates the feasibility of using graphical reasoning techniques to sensor network with an example application in sensor localization. In our work, we propose a graphical model namely HMM construct and map that on sensor network for recognizing ADLs. Each sensor maintains only a small set of probabilities and through communication with the other sensors; it can reach a decision about user activities.

3 Distributed Activity Recognition

In wireless sensor network environment, sensors broadcast their states or the sensed values. The broadcasts are received by the cluster heads and all other sensor nodes in the broadcast range [11]. So, each sensor can capture local data, provided by its neighboring sensor, to do the local computation. We provide a mechanism to process the local data in each sensor with the help of a Hidden Markov Model (HMM) like construct for each activity. We use HMM to capture the temporal relationship among the sensor values which are essential for detecting ADLs that the user performs for certain time duration. We name this as Reconfigurable HMM as we will discuss in the next subsection.

3.1 Reconfigurable HMM

An HMM is defined by a set of states with interconnections between them. A set of prior probabilities; output or emission probabilities for the states and a set of state transition probabilities are defined.

In our approach, each sensor is considered as a state in the HMM. For each activity, a separate HMM is defined. So, prior probability for each state in the HMM is actually the prior probability of the respective sensor being activated during the activity. A state transition probability is the conditional probability of one sensor being activated given another sensor activated before within.

Sensors are activated and deactivated while the user is performing some ADL. If we take the samples of the sensor activations periodically, a sequence of activations can be obtained. The sequence of sensor activations may indicate what the user is actually doing. To capture the sequence information, we construct a virtual HMM with the sensors activated in turns (Figure 2). We omit the state transition links among the sensors in the same sample period because those sensors infer concurrency, not sequence.

After constructing the HMM, the likelihood is calculated for each activity by evaluating their respective HMMs and the highest likely activity is taken. To overcome fluctuation, a decision is given only if the current activity is matched with the previously inferred activity.

The HMM is reconfigurable as the network changes based on the sensors present in the last few time sequences. The worst case is that all the sensors deployed stays within the range and become activated. Keeping the worst case in mind, all the non-zero conditional probabilities are stored. Still the conditional probability table size is not big, unlike one may presume. This is because, ADLs are usually performed in a constrained physical area and only the sensors of that physical area are activated. At this stage we are ready to introduce some terminologies:

$P_a(S_i)$ =Prior probability of sensor i for activity 'a'

$P_a(S_j \mid S_i)$ = conditional probability of S_j given S_i for activity 'a'

$P_a(O_{oi})$ =Probability of output O_o from sensor i for activity 'a'

S_i^t = HMM state constructed with sensor i at time t

$P_a(S_i^t)$ =Calculated probability (usually maximum likelihood) of S_i^t for activity 'a'

$P_a(S_j^t \mid S_i^{t-1})$ =State transition probability of S_j^t given S_i^{t-1} $= P_a(S_j \mid S_i)$

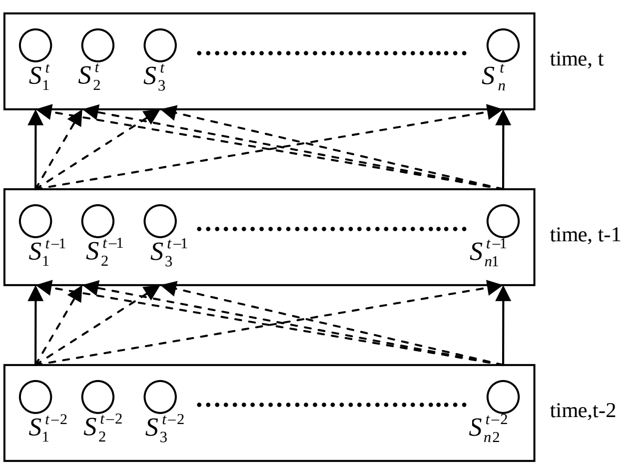

Figure 2. HMM constructed with the sensors in the last three time sequences for any particular activity. By box and arrows we indicate that all states in time (l-1) are connected with all the sensors in time l.

Next we describe our belief propagation algorithms using the terminologies given above:

3.2 Calculating the maximum likelihood

After constructing the HMMs for each activity, we calculate the likelihood for them. We use a maximum likelihood algorithm on the HMM construct, a similar algorithm found in standard message passing literature [10], [13]. Our algorithm also takes the notion of Forward algorithm of HMM [12], where sum operation is replaced by max operation. Each state calculates the maximum likelihood using the formula below:

$$P_a(S_i^l) = P_a(S_i) P_a(O_{oi}), \qquad \text{l=last sequence}$$
$$P_a(S_i^l) = P_a(O_{oi}) \max_c [P_a(S_i^l \mid S_c^{l-1}) P_a(S_c^{l-1})], \quad \text{otherwise} \qquad \ldots\ldots\ldots (1)$$

As the term 'maximum' is distributive in nature, it is possible to calculate the global maximum likelihood from local maximum at each state as discussed in the next section.

3.3 Complexity of the Belief Propagation Algorithms

Suppose that we want to calculate the likelihood of an activity with sensor sequences of length T. So, all the sensors activated within the last T time sequences (taking current time as first) will be involved. First, sensors activated in last T-th time sequence, should calculate their likelihoods using first portion of equation 1 and broadcast the results. Sensors activated in the last (T-1)-th time sequence, should use the likelihoods broadcasted, to calculate their own likelihoods using second portion of the equations 1. This process should continue until the currently activated sensors calculate their likelihood values. So, after a round of (T-1) message passing by the previously activated sensors within last T time sequences, the currently activated sensors can achieve the likelihood of T-sequence. So, the time complexity of each sensor for each activity is Linear with the number of sensors activated within its communication range; communication overhead is the order of sequence length T; space complexity is distributed through the sensors, each sensor containing only the conditional probabilities given all other sensors in the range, in the worst case scenario.

However, the numbers of messages to be sent i.e. the communication overhead can be optimized further and can be reduced to half. This is because, while calculating likelihood of a sensor sequence of length T, a sensor can also calculate the likelihood of its subsequences and broadcast those in a single message. The prerequisite for this is that, sensors activated in previous i-th time sequence (taking current time as first) send the likelihoods of (i+1)-sequence, in addition to sending likelihoods of a sequence of length T-i+1. This advanced calculation can save communication overhead by half. The computational complexity remains the same, as each sensor calculates the likelihoods in advance.

So, the achievement of our approach is using low computation capable sensors for detecting the activities. We propose an in-network processing technique where each sensor contributes to the computation, space and communication required for the task of recognizing ADLs.

4 Experimental Results

We are using the open data provided by MIT Place Lab [1], [14]. They used 77 on/off state sensors in a one person apartment The data were collected in a central place and were labeled using Experience Sampling Method (ESM) [1]. The training set is small with noises as seen from their recognition results [1]. However, the result could further be improved through the use of temporal features constructed from sensor activation sequence, as indicated in [1].

The purpose of using MIT data in our work is to show the effectiveness of our model in a classification problem, where temporal relationship among the features is necessary.

We assume that all the sensors are within their communication ranges and receive the broadcasts from all other sensors. We also assume that all the sensors are time synchronized and broadcast and receive their states periodically all at the same time. We assume this ideal environment to show the improvement of recognition accuracy of ADLs. We have shown the time and computational complexity of our model earlier in section 3.3 analytically.

We use the text data from [14], take samples of sensor values at every 5 seconds and feed through our simulator, a multithreaded program written in java. Each thread represents a sensor. The java program reads the sensor value samples of a particular time instance, passes the values to the threads, i.e. the sensors. The threads communicate with each other and after one synchronized message passing; each thread can calculate the total or maximum likelihood of an activity. The likelihoods are then passed to the program, which acts as a cluster head, to calculate the sum total or global maximum likelihood.

Our simulation program actually emulates the user activities recorded in the data file and provides a continuous sensor activation sequence needed for our algorithm. Our program runs in two phrases. First, we learn the prior and conditional probabilities for the sensors based on this data by frequency counting. In the second phrase, we emulate the ADLs, reproduce the sensor sequences and recognize the activities using algorithm given by equation 1. As the sensors are just on/off sensors, a sensor is considered activated only when it is in on state. So, the only output in active state is on state and the probability of on state is equal to the prior probability of the sensor node, i.e. $P_a(O_{oi}) = P_a(ON_i) = P_a(S_i)$. So, $P_a(O_{oi})$ is replaced by $P_a(S_i)$ in equation 1.

We also filter the data set to reduce the noise. For any activity recognition problem, it is desirable that the classifier recognizes the activity while it is being performed. In our algorithm, the classifier reaches such a state after passing T samples, each taken in 5 seconds interval. However, doing this, reduces the number of samples for large T. So, we relax this constraint and pass a decision for a sample whose previous two samples also have the same labeling.

We have found that, increasing the sequence length increases the recognition accuracy up to a certain limit. Using maximum likelihood belief propagation algorithm as given in equation 1, we have found average recognition accuracy as given in table 1, where 5 is the optimal sequence length. We have tested our algorithm on the sensor data sampled in 1 second and 30 second intervals where we got the highest average accuracy for sequence length 5 (accuracy 47%) and 4 (53%) respectively. We use five fold cross validation on our results.

So, if we run our algorithm on a setup for recognizing the ADLs in [14], we should use a sequence length of 5. The number of messages to be passed will be only 3 and the calculation will be in order of 5, per sensor; which are quite possible in real-time.

Table 1: Average recognition accuracy for ADL data from MIT [14], using maximum likelihood algorithm in equation 1. The accuracy is shown against sensor sequence length.

Seq Len / Accuracy	2	3	4	***5***	6	7	8
	40	48	52	***54***	51	48	45

Table 2 presents the average accuracy results for all the activities given in [14] with a sequence length 5 and sensor values sampled at an interval of 5 seconds.

Table 2: Activity wise recognition accuracy of our algorithm (sequence length 5 and sampling interval 5 seconds)

Activity	Number of Samples	Accuracy in Percentage (%)
Bathing	213	69.95
Toileting	232	56.03
Going_out_to_work	1	100.0
Preparing_lunch	362	35.63
Preparing_dinner	98	31.63
Preparing_breakfast	58	44.96
Dressing	129	69.76
Grooming	260	58.07
Preparing_a_snack	63	58.73
Preparing_a_beverage	60	70.0
Washing_dishes	60	76.66
Doing_laundry	150	61.33
Cleaning	140	18.57
Putting_away_dishes	48	81.25
Washing_hands	4	100.0
Putting_away_groceries	31	74.19
Other	16	93.75
Watching TV	35	48.57

Going_out_for_entertainment	5	80
Lawnwork	39	97.43
Putting_away_laundry	2	0
Average Accuracy		54.02%

We believe that the medium accuracy (54%) of our result is because of the small training set and noise, which is also indicated in [1]. We looked at the confusion matrix of the ADLs and based on that merged a number of activities to prove our claim. We found that meal preparation activities contributed more to the confusion and merging them further improved the accuracy as shown in Table 3:

Table 3: Activity wise recognition accuracy , meal preparation activities treated as a single one

Activity	Accuracy in Percentage (%)
Bathing	71.04
Toileting	56.47
Going_out_to_work	25.0
Meal preparation (Preparing_lunch, Preparing_breakfast, Preparing_dinner, Cleaning, Washing_dishes, Preparing_a_snack, Putting_away_dishes, Putting_away_groceries)	65.21
Dressing	72.44
Grooming	59.02
Preparing_a_beverage	75.0
Doing_laundry	62.0
Washing_hands	100
Other	88.23
Watching TV	60.86
Going_out_for_entertainment	80.0
Lawnwork	97.43
Putting_away_laundry	0
Average Accuracy	65.16%

We also compared our result with the result of MIT [1]. Our algorithm gives 73.47% of accuracy compared their highest accuracy, 27%. Table 4 shows the activity wise performance analysis. The further improvement of our results on a restricted set of activities, having many samples, and clearer boundary, proves the effectiveness our algorithm.

Table 4: Comparison of our result with MIT on the same set of activities

Activity	MIT Result [1] (in %)	Accuracy of our Algorithm [equation 1]
Preparing lunch	25	74.52
Toileting	27	62.77
Preparing breakfast	8	67.87
Bathing	25	63.20
Dressing	7	85.18
Grooming	26	77.63
Preparing a beverage	7	87.14
Doing Laundry	9	84.04

5 Application of our Algorithm

The algorithm we devised is suitable for Smart Collaborative Object (SCO) Environment. By SCOs we understand everyday items such as chairs, books, or medicine cabinets that are augmented with active sensor based computing platforms. Hence, Smart Objects can perceive their environment through sensors, collect information about the context of a nearby user, and collaborate with other objects in their vicinity by means of wireless communication technologies.

Smart objects can report user's state or its own state. As examples, chair can distinguish user sitting, leaning, sitting leg crossed; computer can detect whether the user is working, not; medicine cabinet can infer its own states. Smart Objects are heterogeneous types and each has its own inference algorithm. They can be trained individually and the algorithm can be ported to another object of the same kind.

In a daily living environment objects usually are organized in proximity to each other in the living environment. So, Smart Objects stay in their mutual communication range, as we may expect. We also mentioned earlier that daily activities are usually performed in a physically constrained location, such as in a single room and a sequence of object usage history in that location can indicate the activity being performed. So, our algorithm fits in this type of environment.

Smart Objects can form ad-hoc groups by collaboration based on proximity or type or according to some predetermined guidelines. Inference algorithm, especially the parameters (prior and conditional probabilities) can be downloaded from a server. The algorithms and parameters can be obtained beforehand by forming a scenario and gathering the labeled data. The parameters once trained, can be reused in the same scenario. Super group can be formed by using several subgroups. The training can be done hierarchically in the same way using the output of subgroups. Hence the training becomes tractable and parameters become reusable.

6 Conclusion and Future works

We provide a distributed probabilistic model for recognizing Activities of Daily Living. We prove the distributed nature of our algorithm analytically and show the accuracy by using a small data set collected from a home setting. Given enough data, i.e. the correct probabilities, the model should recognize the activities perfectly. However, getting enough labeled data is very costly and researches are still finding ways. Our next task will be to investigate how to incorporate online learning into our model and learn the probabilities incrementally over time. Consequently, we will implement our prototype in a Smart Collaborative Object (SCO) environment.

7 Acknowledgements

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)"(IITA-2008-(C1090-0801-0002)) and by the MIC(Ministry of Information and Communication), Korea, Under the ITFSIP(IT Foreign Specialist Inviting Program) supervised by the IITA(Institute of Information Technology Advancement, 과제관리번호 : C1012-0801-0003) and by the Ministry of Education and Human Resources Development (MOE), the Ministry of Commerce, Industry and Energy (MOCIE) and the Ministry of Labor (MOLAB) through the fostering project of the Lab of Excellency.

8 References

[1] E. Tapia, S. Intille, and K. Larson. Activity Recognition in the home setting using simple and ubiquitous sensors. Proceedings of PERVASIVE 2004, vol. LNCS 3001, A. Ferscha and F. Mattern, Eds. Berlin Heidelberg: Springer-Verlag, 2004, pp. 158-175.

[2] Donald J. Patterson, Lin Liao, Dieter Fox, and Henry Kautz. Inferring High-Level Behavior from Low-Level Sensors. Ubicomp, 2003.

[3] S. S. Intille, K. Larson, E. Munguia Tapia, J. Beaudin, P. Kaushik, J. Nawyn, and R. Rockinson. Using a live-in laboratory for ubiquitous computing research. In Proceedings of PERVASIVE 2006 , vol. LNCS 3968, K. P. Fishkin, B. Schiele, P. Nixon, and A. Quigley, Eds. Berlin Heidelberg: Springer-Verlag, 2006, pp. 349-365.

[4] Sebastian Luhr, Hung H. Bui, Svetha Venkatesh, Geoff A. W. West. Recognition of Human Activity through Hierarchical Stochastic Learning. First IEEE International Conference on Pervasive Computing and Communications (PerCom'03).

[5] Christopher R. Wren and Emmanuel Munguia Tapia. Toward Scalable Activity Recognition for Sensor Networks. In Proceedings of The Second International Workshop in Location and Context-Awareness (LoCA '06), pp. 168-185, Springer vol. 3987 / 2006

[6] Sangho Park, Jake K. Aggarwal: A hierarchical Bayesian network for event recognition of human actions and interactions. Multimedia Syst. 10(2): 164-179 (2004).

[7] Jeff Bilmes. What HMM Can Do. Technical Report of University of Washington, Department of Electrical Engineering, Number UWEETR-2002-0003. http://www.ssli.ee.washington.edu/people/bilmes/mypapers/whathmmscannotdo04.pdf

[8] E. Munguia Tapia. Activity Recognition in the Home Setting Using Simple and Ubiquitous Sensors. M.S. Thesis Media Arts and Sciences, Massachusetts Institute of Technology, September 2003.

[9] Donald J. Patterson, Dieter Fox, Henry Kautz, Matthai Philipose. Fine-Grained Activity Recognition by Aggregating Abstract Object Usage. In the proceedings of the 2005 Ninth IEEE International Symposium on Wearable Computers (ISWC'05).

[10] Alexander T. Ihler. Inference in Sensor Networks: Graphical Models and Particle Models. Ph. D Thesis, MIT, 2005.

[11] Xiao Renyi, Wu Guozheng. A Survey on Routing in Wireless Sensor Networks. Progress in Natural Science, Vol. 17, No. 3, March 2007, pp. 261 – 269.

[12] Rabiner L. R., Juang B. H. An introduction to hidden Markov models. IEEE ASSP Magazine: 4-15, Jan 1986

[13] Christopher M. Bishop. Chapter 8: Graphical Models, Pattern Recognition and Machine Learning, Springer 2006.

[14] MIT Activity Recognition data: http://courses.media.mit.edu/2004fall/mas622j/04.projects/home/