Conditional Random Fields for Activity Recognition in Smart Environments

Ehsan Nazerfard
Washington State University, Pullman
WA 99164, USA
nazerfard@eecs.wsu.edu

Barnan Das
Washington State University, Pullman
WA 99164, USA
barnandas@wsu.edu

Lawrence B. Holder
Washington State University, Pullman
WA 99164, USA
holder@wsu.edu

Diane J. Cook
Washington State University, Pullman
WA 99164, USA
cook@eecs.wsu.edu

ABSTRACT

One of the most common functions of smart environments is to monitor and assist older adults with their activities of daily living. Activity recognition is a key component in this application. It is essentially a temporal classification problem which has been modeled in the past by naïve Bayes classifiers and hidden Markov models (HMMs). In this paper, we describe the use of another probabilistic model: Conditional Random Fields (CRFs), which is currently gaining popularity for its remarkable performance for activity recognition. Our focus is on using CRFs to recognize activities performed by an inhabitant in a smart home environment and our goal is to validate the claim of its superior performance by comparing CRFs with HMMs using data collected in a real smart home.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: ARTIFICIAL INTELLIGENCE – *Machine Learning*; H.4.m [**Information Systems**]: APPLICATIONS – *Miscellaneous*.

General Terms

Performance, Experimentation.

Keywords

Activity Recognition, Conditional Random Fields, Machine Learning, Smart Environments, Health Assistance.

1. INTRODUCTION

There is growing interest in development of smart environments which are capable to reason about inhabitants in order to provide health monitoring and health assistance. This technology can be used to monitor the activities that inhabitants perform in their everyday settings and to remotely assess their functional well-being. The need for development of such technologies is underscored by the aging of the population, the cost of formal health care, and the importance that individuals place on remaining independent in their own homes. One of the most important steps toward monitoring the functional health of a smart environment resident is to recognize the activities s/he usually performs in the environment. The aim of activity recognition in a smart environment setting is to recognize Activities of Daily Living (ADLs) [1] that people perform in their homes. Activity recognition is essentially a sequence classification problem. Observation data sequences obtained by various sensors are annotated with corresponding activity labels and are used to train different prediction models. Over the past few years, there has been an upsurge of innovative sensor technologies and recognition algorithms for this area from different research groups. The Easyliving project [2] uses the images from two sets of color stereo cameras to track multiple people moving through a space. The eWatch project [3] proposes a new system based on eWatch for activity recognition and monitoring. The Georgia Tech Aware Home [4] identifies people based on pressure sensors embedded into the smart floor in strategic locations. The CASAS project [5] designed and assessed several algorithms that built probabilistic models of activities and used them to recognize activities in complex situations.

Models used for the purpose of activity recognition can be classified as probabilistic [6, 7], logic based [8] or hand-crafted [9]. Probabilistic models are the most popular approaches for this task, because sensor readings are usually noisy and activities are commonly performed in a non-deterministic way. Several probabilistic models have been proposed to model the sequence classification problem of activity recognition and among these, hidden Markov models (HMMs) [10] and more recently Conditional Random Fields (CRFs) [11] are the most common ones. HMMs have been used frequently for activity recognition [12], while CRFs have been proposed originally for the context of natural language processing. CRFs are expanding to other areas including activity recognition [13], although they have typically been dependent on costly accelerometer and RFID sensors.

Studying CRFs[1] and their application to the field of activity recognition in our smart home is the main focus of this paper. In addition, we compare HMMs and CRFs for the purpose of activity recognition in smart home environment.

The remaining of this paper is organized as follows: In section 2, we give a detailed description of the experimental setup, an effective procedure to collect and annotate data. In sections 3 and 4, we discuss about HMMs and CRFs respectively. The more emphasis would be on CRF portion. In section 5, we present the experimental results. Finally in section 6, we conclude the paper and present the future work.

IHI'10, November 11–12, 2010, Arlington, Virginia, USA.

[1] In this paper we only consider linear-chain CRFs.

2. EXPERIMENTAL SETUP

In this section, we describe how data is collected in our smart home testbed, how it is annotated and which features are extracted from collected data.

2.1 Data Collection

The smart home environment testbed that we are using to collect data is 3 story house located in St. Maries, Idaho. The residents in the home were a man, a woman, and a pet. The couple's children also regularly visited the home. The sensor events are generated from motion sensors (these sensor IDs begin with "M") and temperature sensors (these sensor IDs begin with "T"). Figure 1 shows the motion/temperature sensor layout of the smart home. To track people's mobility, we use motion sensors placed on the ceilings and walls. They allow tracking of the people moving across the space. In addition, the testbed also includes custom-built analog sensors to provide temperature readings and hot water, cold water and stove burner use. A sensor network that was designed by our research group captures all sensor events and stores them in a database.

Each sensor data gathered in our study is expressed by four fields which are Data, Time, Sensor ID and Sensor Value. Table 1 provides a sample of these sensor events.

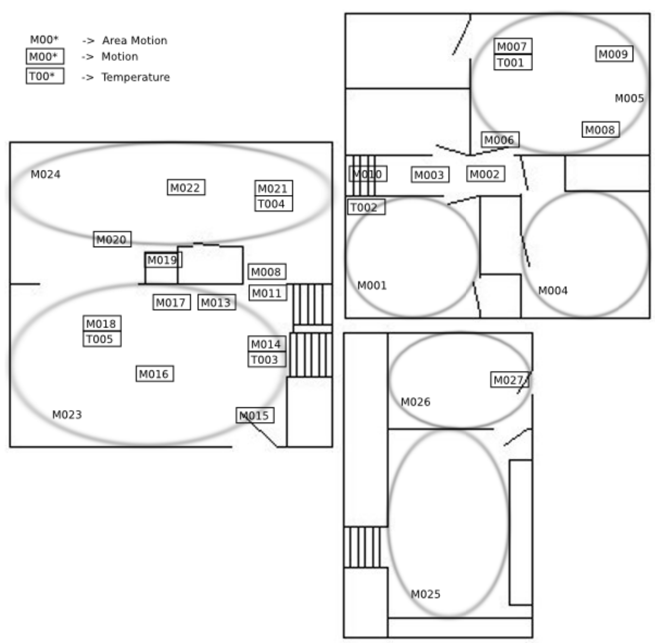

Figure 1. Two bedroom apartment used for experiments.

Table 1. Sample of sensor events used for our study

Date	Time	Sensor ID	Sensor Value
2009-06-10	07:09:57	M013	ON
2009-06-10	07:09:59	M013	OFF
2009-06-10	07:10:08	T003	17.5
2009-06-10	07:10:23	T003	17
2009-06-10	07:10:36	M013	ON

2.2 Data Annotation

After collecting data from a smart home environment, we need to annotate sensor data for activity recognition based on people's activities. Because the annotated data will be used to train the probabilistic models, the quality of the annotated data is very important for the performance of the learning algorithms. A large number of sensor data events will be generated in smart home environments. Without a visualizer tool, it is difficult for researchers and users to interpret raw data as the residents' activities [14].

To improve the quality of the annotated data, we rely upon an open source Python visualizer designed as part of the CASAS project, called PyViz, to visualize the sensor events. Figure 2 shows the user interface of PyViz for the CASAS project. PyViz can display events in real-time or in playback mode from a captured file of sensor event readings. Furthermore, we also built an annotation visualizer to visualize the resident's activities through the annotated file.

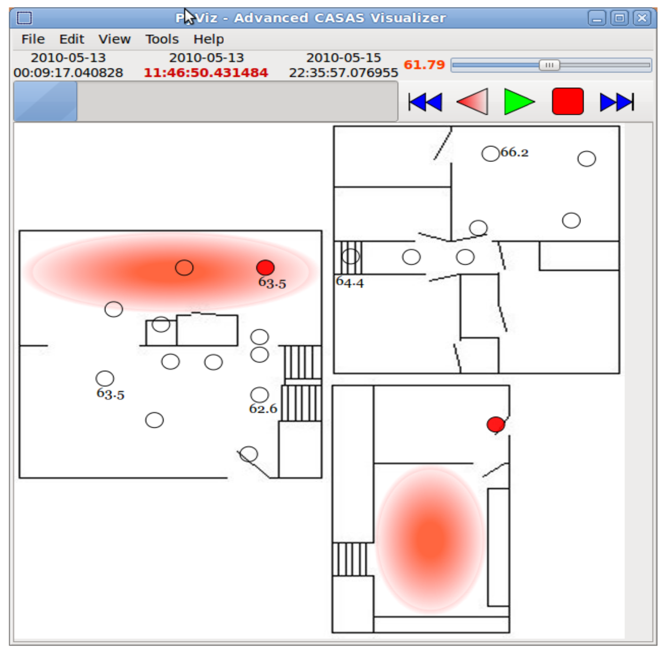

Figure 2. PyViz visualizer for data annotation.

With the help of PyViz, activity labels are optionally added at the end of each sensor event, marking the status of the activity as shown in Table 2. For our experiment, we selected eight activities that the two volunteer participants regularly perform in the smart home environment to classify activities. These activities are as follows:

1. Bed to toilet
2. Breakfast
3. Sleeping
4. Computer work
5. Dining
6. Lunch
7. Night wandering
8. Taking medicine

Table 2. Annotated data format

Date	Time	Sensor ID	Sensor Value	Label
2009-06-10	06:27:30	M023	ON	Breakfast begin
2009-06-10	06:27:30	M016	ON	
2009-06-10	06:27:32	M014	OFF	
2009-06-10	06:27:35	M018	ON	
…	…	…	…	
2009-06-10	06:33:24	M011	OFF	Breakfast end

2.3 Feature Extraction

The features we use to classify the activities are either generated directly from a single event or by considering a set of sensor events. The following is the list and short description of features we are considering for each event:

1. **Sensor:** a logical label identifying the involved sensor.
2. **Time of day:** a discretized value of the time that event occurs. The values are binned into the following hour ranges: 0-5, 5-10, 10-15, 15-20, 20-24.
3. **Day of week:** an integer value representing the day of the week.
4. **Previous activity:** the activity that occurred immediately before the current activity.
5. **Activity length:** the length of the activity in terms of the number of generated sensor events.

To provide real training data for recognizing activities, we have collected data while two residents were living in the smart apartment. Our training data was gathered over a period of several months and more than 100,000 sensor events were generated for our dataset[2].

3. HIDDEN MARKOV MODELS

HMMs are generative models which are used to generate a sequence of hidden states from observable sequences. Due to its rich mathematical structure, HMM forms the theoretical basis of a wide range of applications. Originally proposed by Rabiner et al [10] as an application for signal processing, HMM has gone a long way in the last three decades and found strong applications in speech recognition [15], cryptanalysis, machine translation and bioinformatics.

In the application of activity recognition, hidden states and observations correspond to activity labels and sensor data (features) respectively. Therefore, given an input sequence of sensor event observations, the goal is to find the most likely sequence of hidden states, or activities, which could have generated the observed event sequence.

HMMs represent the joint probability distribution *P(X,Y)*, where *X* refers to an observation sequence and *Y* refers to a label sequence. A typical graphical model of an HMM is represented in Figure 3. To calculate the joint distribution, HMM considers all possible observation sequences. Since it is infeasible in terms of the complexity, HMM assumes that the observations are independent from each other but dependent only on the corresponding label.

[2] Available at http://ailab.eecs.wsu.edu/casas/datasets.html

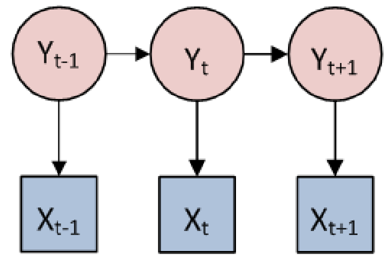

Figure 3. Graphical model of an HMM. Circles correspond to the label (Y) and squares correspond to the observations.

4. CONDITIONAL RANDOM FIELDS

CRFs are probabilistic models which were originally proposed in natural language processing [16], however they are recently applied in wide area of applications including activity recognition [13]. They combine advantages of maximum entropy Markov models (MEMM) without suffering from label bias problem [11].

CRFs are discriminative models, conditioning the probabilities to the observation sequences. They avoid computing the probabilities for every possible observation sequence. Rather than relying on joint probabilities *P(X,Y)*, CRFs specify the probability of possible label sequences given the observation *P(Y|X)*. A typical graphical model of a CRF is depicted in Figure 4, where *X* and *Y* refer to observation and label sequences respectively.

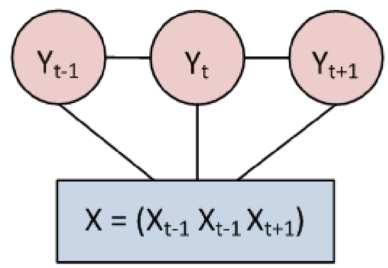

Figure 4. Graphical model of a CRF. Circles correspond to the label (Y) and squares correspond to the observations.

In a smart home setting it is also very important to consider features that link state transitions in the model directly to the observations. Such features are difficult to represent in an HMM due to the way it factorizes probabilities. But, they can be handled by CRFs.

A CRF can be represented as an undirected graph *G = (V, E)*. The probability distribution of an undirected graph is calculated by factorizing maximal cliques[3] $c \in \mathrm{cliques}(V)$ of the graph. We can build a product of potential functions over the set of graph nodes:

$$P(V) = \frac{1}{Z} \prod_{c \in \mathrm{cliques}(V)} \Psi(c) \quad (1)$$

In case of linear-chain CRF, which is the focus of this paper, the potential of each clique takes the form $\Psi(c) = \exp(w.f(t, y_{t-1}, y_t, X))$. After applying the Markov assumption between label sequences *Y*, the following conditional probability is achieved:

[3] A clique is a fully connected subgraph.

$$P(Y|X) = \frac{1}{Z(X)} \exp\left(w \prod_{t=1}^{T} f(t, y_{t-1}, y_t, X)\right) \quad (2)$$

Where *w* is the weight for the feature (calculated during training), and *Z(X)* is a normalization term which guarantees that the distribution sums to one and is given by:

$$Z(X) = \sum_{Y} \prod_{t=1}^{T} \exp\left(w.f(t, y_{t-1}, y_t, X)\right) \quad (3)$$

The above formula is just for one feature. When extend to multiple features we get the following:

$$P(Y|X) = \frac{1}{Z(X)} \prod_{i=1}^{k} \exp\left(w_i \prod_{t=1}^{T} f_i(t, y_{t-1}, y_t, X)\right) \quad (4)$$

Where *T* is the number of labels and *k* is the number of features.

4.1 Training CRFs

Parameter estimation of CRFs is done mainly by frequency and Bayesian approaches [17] which give rise to two techniques: maximum likelihood estimation (MLE) and maximum a priori estimation (MAP), respectively. Because MLE is most commonly used, we focus on MLE for the purpose of training the labeled training set to estimate the weight vector *w*. Though a number of iterative scaling methods like Improved Iterative Scaling (IIS) [18] and Generalized Iterative Scaling (GIS) [19] have been used for parameter estimation, we consider the numerical optimization approach because the function and its gradient are provided during training.

The maximum likelihood parameter estimation for a CRF representing the conditional probability (equation 5) is the task of estimating the weight vector *w* which becomes more convenient if we maximize the log likelihood (equation 6):

$$P(Y|X) = \frac{1}{Z(X)} \prod_{t=1}^{T} \exp\left(w_i f_i(t, y_{t-1}, y_t, X)\right) \quad (5)$$

$$l(Y|X) = \sum_{t=1}^{T} w_i f_i(t, y_{t-1}, y_t, X) - \log\left(Z(X)\right) \quad (6)$$

As the log likelihood function for a CRF is convex over the entire parameter space, first order methods like gradient ascent or conjugate gradient are directly applicable. Differentiating the log-likelihood with respect to w_i we obtain:

$$\frac{dl}{dw_i} = \sum_{t=1}^{T} f_i(t, y_{t-1}, y_t, X) - \sum_{Y} P(Y|X) f_i(t, y_{t-1}, y_t, X)) \quad (7)$$

Setting this derivative or gradient to zero gives the maximum likelihood solution. It can be inferred that the expected value of the feature from the training set must equal the expected value of the same features under the model. It should also be noted that the gradient computation is essentially an exponential sum over all possible sequences which are usually solved by dynamic programming algorithms like the Forward-Backward [20] algorithm. However, in the case of large state spaces, this might prove to be expensive because of the requirement of many calls to the forward-backward algorithm. In this scenario, the training complexity of CRF is overwhelming and can perform worse than a HMM. As a solution to this problem, a Sparse Forward-Backward technique can be used for fast training of CRFs.

4.2 Building Model

As mentioned in Section 2, we train the CRF model on the basis of annotated data. The features we are using are fed to the algorithm to generate a CRF model. Figure 5 depicts a sample model generated from the training set.

As seen from the Figure 5, there might be a substantial amount of interdependence between features, which typically combine information from more than a single event. Unlike HMM, CRF is capable of handling these dependencies.

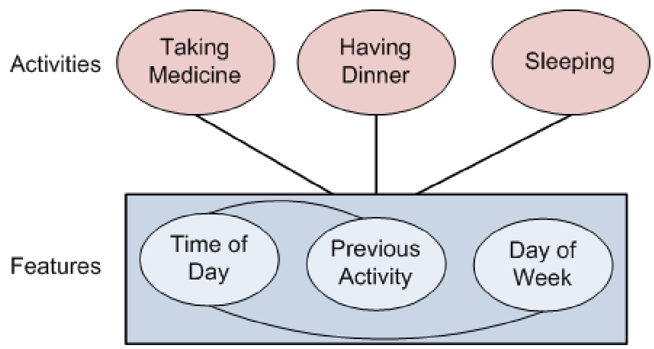

Figure 5. CRF Model representing different activities.

5. EXPERIMENTAL RESULTS

We performed the experiments with a 3 fold cross validation approach for both HMM and CRF and for all the individual activities. As our focus was mainly to learn the behavior of CRF in detail, we compared its accuracy performance for different numbers of training iterations of the algorithm. In Figure 6, we see an interesting pattern of alternating peaks and dips in the graph from the beginning till 10 iterations and again from 19 to 23 iterations. Subsequence increases consistently show an increase in accuracy. As it is trivial that the running time of the algorithm is directly proportional to the number of iterations, we found 30 training iterations to be a good tradeoff between the performance of the algorithm and its running time. Therefore, all the experiments with CRF were performed with 30 training iterations and 3 fold cross validation.

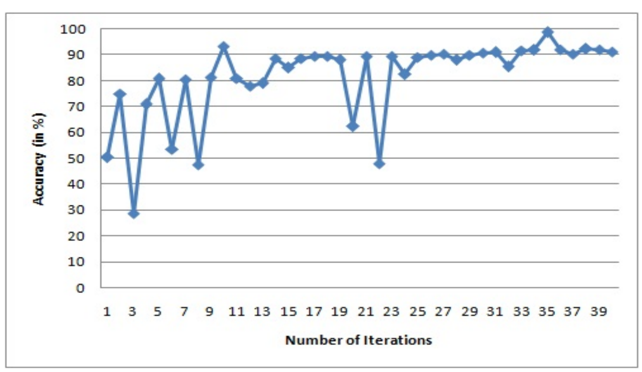

Figure 6. Performance of CRF for different number of iterations.

As per the claims and explanations, CRF has proved to perform better than HMM apart from some specific activities. It gives an average accuracy as high as 91% as compared to HMM which gives 82%. Figure 7 compares the accuracies of HMM and CRF for all the individual activities and Table III, shows their corresponding values. For two activities, Bed to Toilet and Taking Medicine, HMM performs better than CRF. The reason is both of these activities do not involve many different types of sensors and therefore the independence assumption of HMMs works better than the dependency criteria of CRFs.

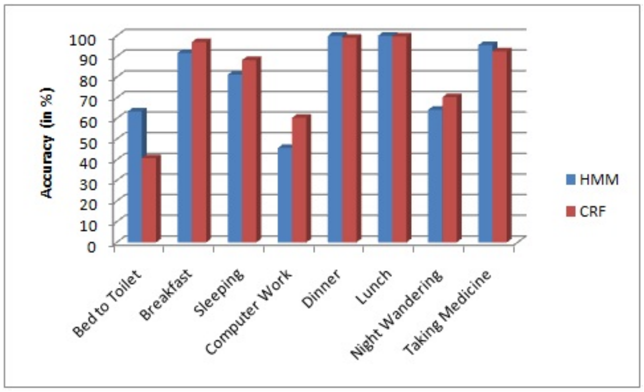

Figure 7. Comparison of accuracy performance by HMM and CRF.

Table 3. Accuracy performance of HMM and CRF

	Accuracy	
Activity	HMM	CRF
Bed to Toilet	63.3333	40.7928
Having Breakfast	91.6667	96.9776
Sleeping	81.1524	88.1997
Computer-Work	45.6522	60.3338
Having Dinner	100	99.0171
Having Lunch	100	99.6237
Night Wandering	64.1791	70.2649
Taking Medicine	95.4545	92.4901

6. CONCLUSION AND FUTURE WORK

In this paper we use conditional random fields for activity recognition, a classical problem in the area of smart environments. We also compare its performance, for recognizing ADLs, to the performance of HMM, another popular probabilistic approach for activity recognition. With our results it can be concluded that CRFs work quite well for data streams. So, it is quite suitable for our projects as the raw sensor data that we collect are in the form of data streams.

In future we want to use CRFs for assessing the completeness of activities. This could be done by recognizing individual steps of an activity that could be compared with statistical inputs from psychologist to determine how well it was completed. Another prospective area in which CRFs could have good application is online activity recognition. Although, attention has been focused in this area, but a little success has been achieved. The reason for this is the incapability of popular machine learning and data mining techniques to handle data streams. As CRFs have the capability to handle this kind of data, it would be a good approach to solve the problem of online activity recognition.

7. REFERENCES

[1] Wadley, V., Okonkwo, O., Crowe, M., Ross-Meadows, L. A. 2007. Mild Cognitive Impairment and everyday function: Evidence of reduced speed in performing instrumental activities of daily living. *American Journal of Geriatric Psychiatry*, 16(5): 416-424.

[2] Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M. and Shafer, S. 2000. Multi-Camera Multi-Person Tracking for Easy Living. *Proc.Third IEEE Intl Workshop Visual Surveillance*, 3-10.

[3] Maurer, U., Smailagic, A., Siewiorek, D., Deisher, M. 2006. Activity recognition and monitoring using multiple sensors on different body positions. *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*.

[4] Orr, R. J. and Abowd, G. D. 2000. The Smart Floor: A Mechanism for Natural User Identification and Tracking. *Proc. Conf. Human Factors in Computing Systems (CHI 00).*

[5] Singla, G., Cook, D. and Schmitter-Edgecombe, M. 2010. Recognizing independent and joint activities among multiple residents in smart environments. *Ambient Intelligence and Humanized Computing Journal*, 1(1): 57-63.

[6] Patterson, D. J., Fox, D., Kautz, H. A., and Philipose, M. 2005. Fine-grained activity recognition by aggregating abstract object usage. *In proc. of ISWC*, 44-51, Osaka, Japan.

[7] Wilson, D. and Atkeson, C. 2005. Simultaneous tracking and activity recognition using many anonymous binary sensors. In *Pervasive Computing*, Third International Conference, 62–79, Munich. Germany.

[8] Landwehr, N., Gutmann, B., Thon, I., Philipose, M. and Raedt, L. 2007. Relational transformation-based tagging for human activity recognition. *Proceedings of the 6th International Workshop on Multi-relational Data Mining (MRDM07)*, 81–92, Warsaw. Poland.

[9] Fogarty, J., Au, C., and Hudson, S. E. 2006. Sensing from the basement: a feasibility study of unobtrusive and low-cost home activity recognition. In *UIST 06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, 91–100, New York, NY. USA.

[10] Rabiner, L. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*. 77(2): 257–286.

[11] Lafferty, J., McCallum, A. and Pereira, F. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proc. *18th International Conference on Machine Learning*. 282–289. Morgan Kaufmann. San Francisco. CA.

[12] Singla, G., Cook, D. and Schmitter-Edgecombe, M. 2009. Tracking activities in complex settings using smart environment technologies. *International Journal of BioSciences, Psychiatry and Technology*. 1(1): 25-35.

[13] Vail, D. L., Veloso, M. M., and Lafferty, J. 2007. Conditional random fields for activity recognition. In *proceedings of the conference on Autonomous Agents and Multi Agent Systems (AAMAS)*, ACM New York, NY, USA.

[14] Kasteren, T., Noulas, A., Englebienne, G. and Krose, B. 2008. Accurate Activity Recognition in a Home Setting. *Proceedings of the Tenth International Conference on Ubiquitous Computing*. Seoul. Korea, 1-9.

[15] Rabiner, L. and Juang, B. H. 1986. An introduction to hidden Markov models. *IEEE Acoust. Speech Signal Proc.*, Mag. 3, 4-16

[16] Wallach, H. 2004. *Conditional random Fields: An Introduction*. Technical Report. University of Pennsylvania CIS.

[17] Wallach, H. 2002. *Efficient Training of Conditional Random Fields*. Master’s Thesis. University of Edinburgh.

[18] Pietra, S. D., Pietra, V. D. and Lafferty, J. 1997. Inducing features of random fields. *IEEE transactions on pattern analysis and machine intelligence*. 19(4):380–393.

[19] Darroch, J. and Ratcliff, D. 1972. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*. 43: 1470–1480.

[20] Sutton, C., Pal, C., and McCallum A. 2005. Sparse forward-backward for fast training of conditional random fields. *NIPS Workshop on Structured Learning for Text and Speech Processing*.