

# Conditional Random Fields versus Hidden Markov Models for activity recognition in temporal sensor data

T.L.M. van Kasteren

A. K. Noulas

B.J.A. Kröse

Intelligent Systems Lab,  
University of Amsterdam ,  
Kruislaan 403, 1098 SJ, Amsterdam , The Netherlands  
{tlmkaste, anoulas, krose}@science.uva.nl

**Keywords:** Activity Recognition, Temporal Sensor Patterns, Conditional Random Fields, Hidden Markov Models

## Abstract

Conditional Random Fields are a discriminative probabilistic model which recently gained popularity in applications that require modeling non-independent observation sequences. In this work, we present the basic advantages of this model over generative models and argue about its suitability in the domain of activity recognition from sensor networks. We present experimental results on a real-world dataset that support this argumentation.

## 1 Introduction

Human activities are a very informative piece of information for many pervasive computing applications. One such application lies in healthcare, where the increasing number of elderly people causes much concern in terms of costs and lack of personnel. A solution is to allow elderly to live independently at home longer. This can be realized by monitoring the activities that elderly perform and provide healthcare personnel with valuable information with regards to the elders wellbeing. Such activities of daily living (ADLs) are a standardized piece of information in healthcare, representing the cognitive and physical capabilities of an elderly person [2]. Typical examples include bathing, cooking and cleaning.

In order to recognize which activities are taking place, a wireless sensor network needs to be installed in the house. Sensor networks consist of a number of easy to install, practically invisible wireless sensor nodes that can run on batteries for a year. They are typically equipped with contact switches, pressure mats and motion detectors to provide binary feedback of, for example, a door being opened, someone sitting on a chair or being in a room. Even if it is trivial to collect this sensor information with modern technol-

ogy, we are still left with the formidable task of inferring the high level person activities from these simple readings. Since human behavior is not deterministic and it is not known beforehand when an activity starts and ends, the latter is a very challenging problem.

Activity recognition has been studied for quite some time, however, most research has been on recognition from video data with a focus on rather simple activities such as walking or running [4]. More recently there has been an increase in research on activity recognition from sensor networks data where more complex activities, such as cooking or bathing, are recognized [10, 5, 12].

Previous approaches to activity recognition from sensor networks have mainly used generative models. In these models, we explicitly state how the observations are generated from the real-world process. The task is then split in two parts: estimating the process parameters from our data (training) and use these parameters to infer the real-world process by looking at the novel sensor readings. Different generative models exist, and we can separate them in two main categories. Generative models that do not take into account temporal patterns in our data, for instance the naive Bayes model [10], and generative models that use these temporal dimension to extract vital information, like for instance different kinds of Hidden Markov Models (HMMs) [5], [12].

These generative models have been used for activity recognition in sensor networks in [10], [5] and [12]. Comparative research has clearly shown the advantage of using generative models that utilize temporal information in our data [11], something well expected considering the nature of our problem.

A drawback of HMMs is that they require multiple assumptions to keep the training and inference parts tractable. These assumptions explicitly ignore possible long term dependencies (Markov assumption) and

non-independent features existing among the observations. In activity recognition for sensor networks, none of these assumptions hold. The complexity and variability of temporal patterns of ADLs imply long term dependencies, while each sensor network has a specific spatial setting, which inevitably lead to dependencies in the sensor firings.

It follows that relaxing those assumptions will lead to a more accurate inference about the ADLs performed in our sensor network. Conditional random fields (CRFs) are a type of discriminative model that allow exactly that, to model long term dependencies and use multiple dependent features in a principled way [3].

In this paper, we investigate the applicability of HMMs and CRFs in the problem of activity recognition from sensor networks. To our best knowledge, this is the first time that conditional models are applied in this field, although they have exhibited excellent results in other domains with similar difficulties [6, 8, 3]. The contribution of our work is twofold. Firstly, we present a theoretical comparison of the two models, focused on the field of activity recognition, that reasons in favor of the conditional models. Secondly, we present the results of extended experiments with real-world data, that support this reasoning.

The remainder of this paper is organized as follows, we first formalize the problem of activity recognition, and describe how HMMs and CRFs tackle the tasks of training and inference. Then, we discuss how they differ and how this differences affect their performance in the task of activity recognition. Finally we present the results acquired through a series of experiments and conclude the paper with a discussion regarding possible applications and future research directions.

## 2 Activity Recognition in Sensor Networks

Our objective is to recognize activities of daily living (ADLs) from sensor readings in a house. To do this, we need to divide our time series data (observations) in time slices of constant length and determine the activity (label) for each slice. We denote the duration of a time slice with  $\Delta t$ . We denote a sensor reading for time  $t$  as  $x_t^i$ , indicating whether sensor  $i$  fired at least once between time  $t$  and time  $t + \Delta t$ , with  $x_t^i \in \{0, 1\}$ . In a house with  $N$  sensors installed, we define a binary observation vector  $\vec{x}_t = (x_t^1, x_t^2, \dots, x_t^N)^T$ . The activity at time slice  $t$  is denoted with  $y_t$  and so formally our task is to find a mapping between a sequence of observations  $\mathbf{x} = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_T\}$  and a sequence of labels  $\mathbf{y} = \{y_1, y_2, \dots, y_T\}$  for a total of  $T$  time steps (fig. 5).

We utilize the framework of probabilistic models

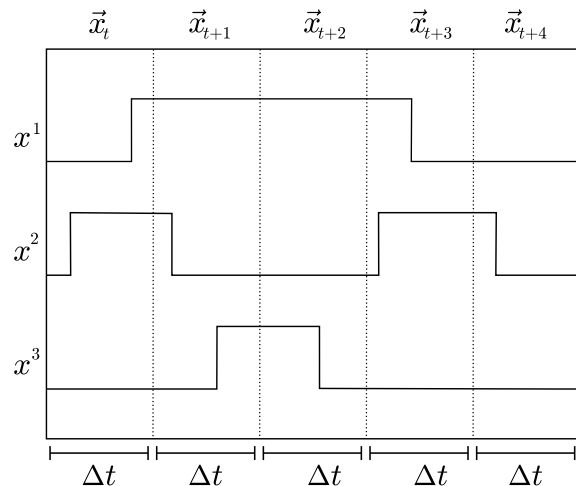


Figure 1: Showing the relation between sensor readings  $x^i$  and time intervals  $\Delta t$ .

to capture this mapping, which requires us to do two things: First, we have to learn the parameters of our probabilistic model from training data. Second, we have to infer which sequence of labels best explains our observations. In this section we will present how these two steps work for both HMMs and CRFs.

### 2.1 Hidden Markov Model

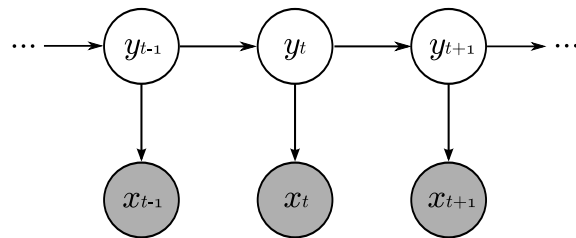


Figure 2: The graphical representation of a HMM. The shaded nodes represent observable variables, while the white nodes represent hidden ones.

The Hidden Markov Model (HMM) is a generative probabilistic model consisting of a hidden variable and an observable variable at each time step (fig. 2). In our case the hidden variable is the ADL performed, and the observable variable is the vector of sensor readings. There are two dependency assumptions that define this model, represented with the directed arrows in the figure.

- The hidden variable at time  $t$ , namely  $y_t$ , depends only on the previous hidden variable  $y_{t-1}$  (*Markov assumption*).
- The observable variable at time  $t$ , namely  $x_t$ , depends only on the hidden variable  $y_t$  at that time slice.

With these assumptions we can specify an HMM using three probability distributions: the distribution over initial states  $p(y_1)$ ; the transition distribution  $p(y_t|y_{t-1})$  representing the probability of going from one state to the next; and the observation distribution  $p(x_t|y_t)$  indicating the probability that observation  $x_t$  was generated by the state  $y_t$ .

Learning the parameters of these distributions is done by maximizing the joint probability  $p(\mathbf{x}, \mathbf{y})$  of the paired observation and label sequences in the training data. We can factorize the joint distribution in terms of the three distributions described above as follows [9]:

$$p(\mathbf{x}, \mathbf{y}) = \prod_{t=1}^T p(y_t | y_{t-1}) p(x_t | y_t) \quad (1)$$

in which we write the distribution over initial states  $p(y_1)$  as  $p(y_1 | y_0)$ , to simplify notation.

Finding the parameters that maximizes this joint probability in our case can be done by frequency counting, since we are dealing with discrete data [7].

Inferring which sequence of labels best explains a new sequence of observations can be performed efficiently using the Viterbi algorithm [7].

What is important for our comparison with CRFs are the following two facts. First, notice that the observation distribution  $p(x_t|y_t)$  is a distribution conditioned on the hidden state, this is why the HMM is a *generative* model. Because of the generative nature of the model we are trying to learn a distribution over the possible observations given the state. However, during inference it is the observation that is given. It would therefore be more intuitive to condition on the observation, rather than the state. Second, keep in mind, parameter learning was done by maximizing the *joint* probability  $p(\mathbf{x}, \mathbf{y})$ .

## 2.2 Conditional Random Fields

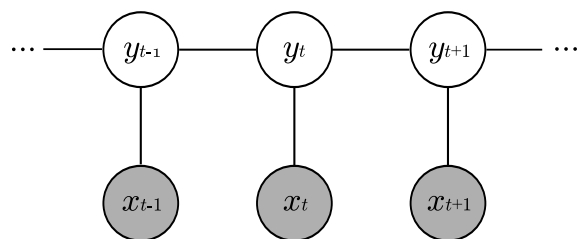


Figure 3: The graphical representation of a linear-chain CRF. The shaded nodes represent observable variables, while the white nodes represent hidden ones.

A Conditional Random Field (CRF) is a discriminative probabilistic model that can come in many different forms. The form that most closely resembles the HMM is known as a linear-chain CRF and is the

model we use in this paper (fig. 3). As the figure shows, the model still consists of a hidden variable and an observable variable at each time step. However, the arrowheads of the edges between the various nodes have disappeared, making this an undirected graphical model. This means that two connected nodes no longer represent a conditional distribution (e.g. a given  $b$ ), but instead we speak of the potential between two connected nodes. This potential is also some sort of correlation measure, but unlike a probability is not restricted to be a value between 0 and 1 [1].

The potential functions that specify the linear-chain CRF are  $\psi(y_t, y_{t-1})$  and  $\psi(y_t, x_t)$ . For clarity of representation these potential functions are written down using a more uniform notation, which allows different forms of CRFs to be expressed using a common formula. In this work, we adopt that notation [9, 1] and therefore define:  $\psi(y_t = i, y_{t-1} = j) = \lambda_{ijk} f_{ijk}(y_t, y_{t-1}, x_t)$  in which the  $\lambda_{ijk}$  is the parameter value (the actual potential) and  $f_{ijk}(y_t, y_{t-1}, x_t)$  is a feature function that in the simplest case returns 1 when  $y_t = i$  and  $y_{t-1} = j$ , and 0 otherwise. By defining the other potential function similarly we have our consistent notation:  $\psi(y_t = i, x_t = k) = \lambda_{ijk} f_{ijk}(y_t, y_{t-1}, x_t)$ , again  $\lambda_{ijk}$  is the parameter value and the feature function now returns 1 when  $y_t = i$  and  $x_t = k$ , and 0 otherwise. The index  $ijk$  is typically replaced by a one-dimensional index, so we can easily sum over all the different potential functions. It should be noted that more complicated feature functions can be used, however, they should always return a scalar [9].

All this, is largely just a notational difference with HMMs, caused by the undirected graphical model representation. The conditional probabilities  $p(x_t|y_t)$  and  $p(y_t|y_{t-1})$  have been replaced by the corresponding potentials. The difference lies in the way we learn the model parameters. In the case of HMMs the parameters are learned by maximizing the *joint* probability distribution  $p(\mathbf{x}, \mathbf{y})$ . The parameters of a CRF are learned by maximizing the *conditional* probability distribution  $p(\mathbf{y} | \mathbf{x})$ . One of the main consequences of this choice, is that while learning the parameters of a CRF we avoid modelling the distribution of the observations,  $p(x)$ . As a result, we can only use CRFs to perform inference (and not to generate data), which is a characteristic of the discriminative models. In ADLs recognition, the only thing we are interested in is classification and therefore CRFs fit our purpose perfectly. The question that remains is how do we get from the potential functions to the conditional probability distribution  $p(\mathbf{y} | \mathbf{x})$ , the formula for this is as follows [9]:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(x)} \exp \left\{ \sum_{k=1}^K \lambda_k f_k(y_t, y_{t-1}, x_t) \right\} \quad (2)$$

where  $Z(x)$  is the normalization function

$$Z(x) = \sum_{\mathbf{y}} \exp \left\{ \sum_{k=1}^K \lambda_k f_k(y_t, y_{t-1}, x_t) \right\} \quad (3)$$

We see here that the formula sums over all the different potential functions. The feature function  $f_k(y_t, y_{t-1}, x_t)$  will return a 0 or 1 depending on the values of the variables and therefore decides whether a potential should be included in the calculation. Finally the sum of potentials is divided by the normalization term  $Z(x)$  which guarantees that the outcome is a probability.

Model parameters can be learned using an iterative gradient method. Particularly successful have been quasi-Newton methods such as BFGS, because they take into account the curvature of the likelihood function. Inference can be performed using slightly altered version of the viterbi algorithm [9].

In conclusion, CRFs have become so popular recently because they exhibited excellent results in time series data analysis. This is due to their potential to model the conditional distribution  $p(\mathbf{y} | \mathbf{x})$  directly. Since we only want to use the model for inference, we do not have to consider the form of  $p(\mathbf{x})$ . This allows us to include all sorts of rich, overlapping features as observations, which in the generative case would either lead to intractable models or hurt accuracy. Simply put, CRFs make independence assumptions among  $\mathbf{y}$ , but not among  $\mathbf{x}$  [9]. Note that both the models proposed here do not take into consideration any distant temporal dependencies. The difference is that the CRF tries to capture the pattern appearing in the feature space of a single time slice.

### 3 Experiments

The objective of our experiments is to compare the performance and evaluate the potential of HMMs and CRFs in activity recognition from sensor networks. We use one artificial and one real dataset. The artificial dataset, section 3.1 confronts the two models with sensor patterns of realistic activities. The advantage is the possibility to have infinite amount of accurately labeled data. The real dataset, described in section 3.2, is more interesting in terms of future potential, however it is of much lower quality. Finally, the details of our experimental setup can be found in section 3.3.

#### 3.1 Simulated Data

Our simulator randomly creates  $n$  number of activities. The first activity is sampled from a uniform dis-

tribution while the consequent activities are sampled from the real distribution estimated through the pre-defined transition probabilities. Each activity is defined by a fixed pattern of sensor firings, which can be set manually. A sensor pattern defines the order in which the different sensors fire. For example, the sensor pattern 1, 2, 3 shows that sensor 1 fired first, then sensor 2 and then sensor 3. A noise parameter defines the chance of a random sensor firing appearing in between two consecutive sensor firings within the pattern. The time and duration at which the sensors fire is sampled from a gamma distribution, which generates times ranging from around 0 to 30 minutes.

For our experiments we configured the simulator to use five sensors and three activities. We generated data for two type of experiments.

The first type consisted of activities that could be distinguished by one of the five sensors not firing in the sensor pattern. For example, one activity would have sensor pattern 1, 2, 3, 4, while another would have 5, 4, 3, 2. Our models should be able to easily capture the distinction between activities from the missing sensor, although the start and end point of an activity remains a challenge.

The second type of experiments consisted of activities in which all five sensors fire, but in a different order. For example, one activity might have sensor pattern 1, 2, 3, 4, 5, while another would have 5, 4, 3, 2, 1. This should pose a real challenge for our models as it is difficult to distinguish activities merely on the order of sensor firings.

#### 3.2 Real World Dataset

The dataset that we used in our experiments consists of sensor readings recorded in the house of a 30-year-old woman, which was originally used in [10]. She lives alone in an one-bedroom apartment where 77 state-change sensors were installed. Sensors were left unattended, collecting data for 14 days in the apartment. Activities were annotated by the woman herself using a PDA device, choosing from a list of 13 different activities. The different activities are: 'Going out to work', 'Toileting', 'Bathing', 'Grooming', 'Dressing', 'Preparing breakfast', 'Preparing lunch', 'Preparing dinner', 'Preparing a snack', 'Preparing a beverage', 'Washing dishes', 'Cleaning' and 'Doing laundry'. An important notice here is that there is an "idle" activity as label of all the time slices for which we have no information. This activity is by far the most common, and makes the dataset particularly hard: because some activities were missed during annotation, training data of the "idle" class might actually belong to one of the others. In order to measure the effect of this dominant class, we perform our experiments on this datasets with the idle activity omitted as well. The full dataset contains a total of 2989

sensor firings labeled annotated with 278 separate instances of activities.

### 3.3 Setup

In our experiments the sensor readings are divided in data segments of length  $\Delta t = 300$  seconds, since this choice gave the optimal results in earlier studies [11]. We separate our data into a test and training set we used a 'leave one day out' approach. In this approach, one full day of sensor readings is used for testing and the remaining days are used for training. In this way, we get inferred labels for the whole dataset by concatenating the results of each day.

We evaluate the performance of our models by two measures, the time slice average and the class average. These measures are defined as follows:

**Time slice:** 
$$\frac{\sum_{n=1}^N [inferred(n)=true(n)]}{N}$$

**Class:** 
$$\frac{1}{C} \sum_{c=1}^C \left\{ \frac{\sum_{n=1}^{N_c} [inferred_c(n)=true_c(n)]}{N_c} \right\}$$

in which  $[a = b]$  is a binary indicator giving 1 when true and 0 when false.  $N$  is the total number of time slices and  $C$  is the number of classes. The subscript  $c$  indicates a subset that belongs to class  $c$ .

Measuring the time-slice accuracy is a typical way of evaluating time-series analysis. However, we also report the class average accuracy, which is a common technique in datasets with a dominant class. In these cases classifying all the test data as the dominant class yields good time-slice accuracy, but no useful output. The class average though will remain low, and therefore be representative of the actual model performance.

Our *raw* sensor representation simply gives a 1 when the sensor is firing and a 0 otherwise (fig. 5a). Next to this raw data as observations we also tried a *change point* representation. In this representation, the sensor gives a 1 when the sensor changes from 0 to 1 or from 1 to 0 and gives a 0 otherwise (fig. 5b). The *change point* representation doesn't contain more data, but rather clusters the feature space in the form of a XOR operator.

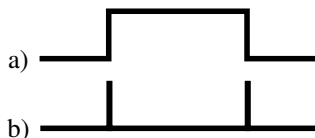


Figure 5: Example of sensor firing showing the a) *raw* and b) *change point* observation representation.

## 4 Results and Discussion

In our experiments on simulated data we tested the accuracy of our models as function of the percent-

Accuracy %	Timeslice	Class
HMM - raw	73.2%	32.9%
CRF - raw	70.4%	22.0%
HMM - changepoint	<b>83.3%</b>	<b>36.4%</b>
CRF - changepoint	82.7%	29.8%

Table 1: Classification accuracy for the HMM and CRF models in the task of time-slice labeling and activity detection for the dataset with idle activity *included*.

Accuracy %	Timeslice	Class
HMM - raw	39.4%	31.6%
CRF - raw	35.7%	28.3%
HMM - changepoint	48.5%	<b>36.1%</b>
CRF - changepoint	<b>49.7%</b>	35.8%

Table 2: Classification accuracy for the HMM and CRF models in the task of time-slice labeling and activity detection for the dataset with idle activity *omitted*.

age of noise in the sensor pattern. In section 4.1 we present the results of those experiments. In section 4.2 we present the results on the real dataset. We discuss these results in section 4.3.

### 4.1 Simulated data

In the simulated data there is no dominant class and therefore we only present the time slice accuracy. The artificial dataset of type 1, contains activities easily differentiable by the absences of a specific sensor. Both models exhibit high accuracy, visible in the left side of figure 4, with more than 70% of the activities correctly identified, even with more than half the sensor reading corresponding to noise. On the other hand, the artificial dataset of type 2 is much harder, and all models perform significantly worse. A final important notice is that the change-point representation does significantly worse than the raw data.

### 4.2 Real World Dataset

In tables 1 and 2 we present compactly our classification accuracy results on real world data, with the idle activity included or not respectively. It is important to notice that in this case, the change-point representation performs significantly better than the raw data. Furthermore, HMMs outperform CRFs when the idle activity is included, and perform approximately the same when the idle activity is omitted.

### 4.3 Discussion

The first important observation is that CRFs perform better in both artificial datasets. Furthermore, in the type 2 dataset this difference is even larger. This is

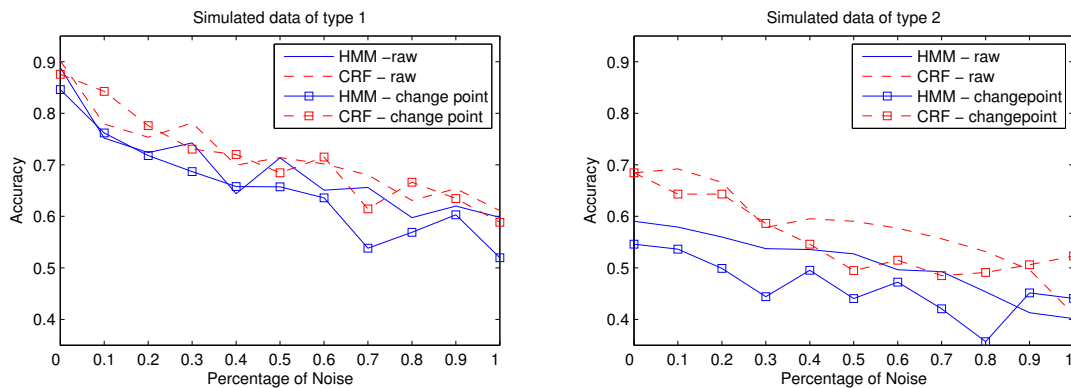


Figure 4: Time slice accuracy for the HMM and CRF models as a function of noise in artificial dataset of type 1 (left) and type 2 (right).

because discriminative models, like CRFS, can concentrate on the differences between similar looking classes.

On the other hand, in the real world dataset HMM performed better. Furthermore, when the idle activity is excluded both models perform equally well. The reason for this is that some activities that were performed were not labeled, and thus considered idle. This poor annotation has a bigger impact on the accuracy of a discriminative model, because during learning classes compete to find the best discriminative fit. While in the generative case the parameters for each class can be considered as learned separately.

## 5 Conclusions

In this paper we investigated the applicability of HMMs and CRFs in activity recognition from sensor networks. Theoretical analysis was in favor of the CRF framework and the best experimental results were also achieved for CRFs. The main conclusion from this work is that the domain of CRFs fits naturally to the problem of detecting activities from sensor networks. The discriminative nature of the model, captures complex dependencies in the observation vector and provides improvement over the results of the typical methods used so far.

## Acknowledgements

This work is part of the Context Awareness in Residence for Elders (CARE) project. The CARE project is partly funded by the Centre for Intelligent Observation Systems (CIOS) which is a collaboration between UvA and TNO, and partly by the EU Integrated Project COGNIRON (The Cognitive Robot Companion). A.K. Noulas was funded by MultimediaN.

## References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science*

*and Statistics)*. Springer, August 2006.

- [2] S. Katz, T.D. Down, H.R. Cash, and et al. Progress in the development of the index of adl. *Gerontologist*, 10:20–30, 1970.
- [3] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.
- [4] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst.*, 104(2):90–126, 2006.
- [5] Donald J. Patterson, Dieter Fox, Henry A. Kautz, and Matthai Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *ISWC*, pages 44–51. IEEE Computer Society, 2005.
- [6] Ariadna Quattoni, Michael Collins, and Trevor Darrell. Conditional random fields for object recognition. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1097–1104. MIT Press, Cambridge, MA, 2005.
- [7] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [8] Cristian Sminchisescu, Atul Kanaujia, Zhiguo Li, and Dimitris Metaxas. Conditional models for contextual human motion recognition. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1808–1815, Washington, DC, USA, 2005. IEEE Computer Society.

- [9] Charles Sutton and Andrew McCallum. *Introduction to Statistical Relational Learning*, chapter 1: An introduction to Conditional Random Fields for Relational Learning, page (Available online). MIT Press, 2006.
- [10] E. Munguia Tapia, S. S. Intille, and K. Larson. Activity recognition in the home setting using simple and ubiquitous sensors. In *PERVASIVE*, 2004.
- [11] T. L. M. van Kasteren and B. J. A. Kröse. Bayesian activity recognition in residence for elders. In *Intelligent Environments*, 2007.
- [12] Daniel Wilson and Chris Atkeson. Simultaneous tracking and activity recognition (star) using many anonymous binary sensors. In *Pervasive*, 2005.