Recognizing Interleaved and Concurrent Activities: A Statistical-Relational Approach

Rim Helaoui, Mathias Niepert and Heiner Stuckenschmidt
KR & KM Research Group
University of Mannheim
Mannheim, Germany
$\{$*rim, mathias, heiner*$\}$*@informatik.uni-mannheim.de*

***Abstract*—A majority of the approaches to activity recognition in sensor environments are either based on manually constructed rules for recognizing activities or lack the ability to incorporate complex temporal dependencies. Furthermore, in many cases, the rather unrealistic assumption is made that the subject carries out only one activity at a time.**

In this paper, we describe the use of Markov logic as a declarative framework for recognizing interleaved and concurrent activities incorporating both input from pervasive light-weight sensor technology and common-sense background knowledge. In particular, we assess its ability to learn statistical-temporal models from training data and to combine these models with background knowledge to improve the overall recognition accuracy. To this end, we propose two Markov logic formulations for inferring the foreground activity as well as each activities' start and end times. We evaluate the approach on an established dataset. where it outperforms state-of-the-art algorithms for activity recognition.

I. Introduction

Bridging the gap between computer models and the physical world is one of the central goals of pervasive computing systems. In order to reach this goal, pervasive systems have been equipped with novel perceptive faculties such as RFID sensor technology opening up novel technological opportunities and application areas. One of these areas, activity recognition in ambient assisted living (AAL) environments, is motivated by the need for IT systems that facilitate reliable and affordable health-care in an aging world population. Through the continuous sensing of a subject's daily routines, personalized care and efficient assistance can be provided at lower costs. This is especially interesting for several common scenarios such as physical and mental rehabilitation or care-taking of elderly patients and patients with cognitive impairments.

The feasibility of such adaptive and proactive support systems depends crucially on novel health-care infrastructure to approach the challenge of autonomously recognizing the daily activities of patients in their home environment. Given a series of sensor observations, the challenge of activity recognition consists in recognizing the particular activities in order to detect and avoid emergency situations. This task is also referred to as plan recognition, goal recognition, intent recognition, and behavior recognition in closely related fields (Hu $et\,al.$ [1]).

Several researchers, like M. Cirillo $et\,al.$ for example, have approached activity recognition by leveraging heterogeneous audio and visual sensor inputs (Cirillo $et\,al.$ [2], Biswas $et\,al.$ [3], Tran and Davis [4]). Such expensive tracking systems, however, are prone to domain-dependent performance and raise several privacy issues. Many of these shortcomings can be avoided by using light-weight sensor data such as the one obtained from RFID sensor networks. The data derived from such sensor networks is used as training data in several current approaches. Most of the state-of-the-art prediction systems rely on data-driven supervised learning paradigms such as conditional random fields and hidden Markov models (Buettner $et\,al.$ [5] and Patterson $et\,al.$ [6]).

On the other hand, recent efforts to explore weakly supervised learning (Stikic and Schiele [7]) as well as unsupervised learning approaches (Tao Gu $et\,al.$ [8]) have been developed. Through such techniques, the authors attempt to overcome the bottleneck of obtaining large amounts of training data. Furthermore, as an alternative to data-driven approaches, Jeon $et\,al.$ [9] have investigated knowledge based algorithms in the context of activity recognition. While data shortage and sparseness is not as problematic for such approaches, they still require a-priori knowledge and have some limitations in the presence of uncertainty. We believe that a combination of both the data-driven and knowledge-based paradigms has the potential to solve a large class of activity recognition problems due to their ability to capture rich and long-term contextual information while also taking advantage of advances in learning and reasoning with probabilistic models.

In this paper, we propose to use Markov logic networks MLN (Richardson and Domingos [10]) as a statistical relational framework for activity recognition in AAL environments. Markov logic incorporates both *hard* logical statements as well as *soft* uncertain evidence with a unifying syntax and semantics. This allows, for example, to integrate existing domain knowledge and, therefore, reduces the amount of training data needed. We also believe that, unlike the majority of the proposed temporal probabilistic models, Markov logic is an excellent framework to integrate rich temporal context without redeveloping a novel model each time. Temporal information is a crucial aspect in human activity recognition (Cirillo $et\,al.$ [2]) and, due to its declarative nature, Markov logic enables the system designer to integrate and test more complex dependencies simply by adding additional formulae.

This becomes even more important, when the user acts under several goals simultaneously. While most approaches to activity recognition assume that the subject carries out one activity at a time, we consider the problem in a more general formulation. Since multitasking is generally an inherent characteristic in real world daily routines as shown by Hao Hu $et\,al.$ [1], we do not presuppose the activities to be strictly sequential.

To summarize, many existing approaches rely on manually designed rules for recognizing activities or lack the ability to take temporal knowledge into account. Furthermore, they usually simplify the problem by attempting to only recognize sequential activities which is usually an unrealistic assumption. The purpose of our work is to investigate the use of Markov logic for recognizing interleaved activities based on input from both pervasive light-weight sensor technology and common-sense knowledge bases. In particular, we want to evaluate its ability to capture qualitative temporal relations and background knowledge for improving the recognition accuracy (Helaoui [11]).

The paper is organized as follows. The next section discusses related work. Section 3 describes the problem statement. In section 4, we provide an overview about Markov Logic Networks. The proposed activity recognition framework is described in section 5. Finally, we present our experiments, discuss their results and outline our future work in the last two sections.

II. Related Work

We consider the general problem of recognizing activities that are not necessarily sequential and uninterrupted. There are only few approaches that make this less restrictive and more challenging assumption. These include Patterson $et\,al.$ [6], who collected sensor data from the morning routines of users performing eleven interleaving activities. These activities share a large number of sensor-tagged objects the subjects can interact with. In this setting, HMMs with one state for each activity are employed as prediction algorithm. Furthermore, the authors consider increasingly more complex models.

The improvements of the more involved methods, however, were only modest suggesting that HMMs are more suitable for purely sequential activities (Kim $et\,al.$ [12]). Since CRFs have shown similar limitations in recognizing interleaved activities (Kim $et\,al.$ [12]), a more sophisticated variant, SCCRF, has been proposed by Hu and Yang [13]. It can model long distance temporal dependencies leveraging so-called *skip edges*. The authors also use correlation graphs to model concurrent activities and formulate the recognition problem as an instance of quadratic programming. However, SCCRF potential functions pose a computationally expensive inference problem especially when a large numbers of skip edges is involved (McCallum and Sutton [14]). Furthermore, to prevent the recognition accuracy from deteriorating, every partial model of the interleaved activities has to be observed during the training phase. Hence, SCCRFs require a large amount of training data as there are numerous ways to interrupt and resume an ongoing activity. This also is the case for (Modayil $et\,al.$ [15]), where a variant of HMMs is proposed. The so-called interleaved HMMs (IHMM) model the activities in the context of the last object used and process event counts to compute the transition probabilities.

In Gu $et\,al.$ [8] RFID tagged objects and wearable accelerometers are deployed and emerging patterns (EP) with sliding windows are used to address the recognition problem. This relies on calculating complex activity scores based on mined activity-feature sets as well as correlation scores between the activities. The activity with the highest score is then chosen. However, this approach is prone to limitations since a sliding time window might exclude some of the distinguishing features. This imposes the use of a segmentation algorithm to improve the results. In addition, the approach is not able to capture long-term temporal dependencies.

Each of these approaches is data-driven and, therefore, lacking the ability to integrate common-sense background knowledge which we believe to be an important faculty of any activity recognition system. Hence, it is at best cumbersome to add and acquire further contextual information to these model. Most related to our work, Riboni and Bettini [16] have approached the recognition problem from a knowledge based perspective. They combine ontological reasoning and multi-class logistic regression (MLR) for probabilistic activity recognition. The core component is an ontology representing activities, symbolic locations, persons, and time granularities. The ontology is queried to infer the subject's context reducing potential mis-predictions by the probabilistic algorithms. Although similar to our approach, the approach does not address the temporal relationships between activities and is restricted to purely sequential activities.

Similarly, Biswas $et\,al.$ [3] and Tran and Davis [4] presented a statistical relational method for activity recognition. Particularly, they use MLNs and dynamic MLNs (DMLN) to solve this task while also incorporating common-sense knowledge as proposed in our work. However, their algorithms are designed with visual activity recognition in mind and take video data as main input. Moreover, they address a rather limited temporal context and only simple *atomic* sequential activities such as shaking hands (Tran and

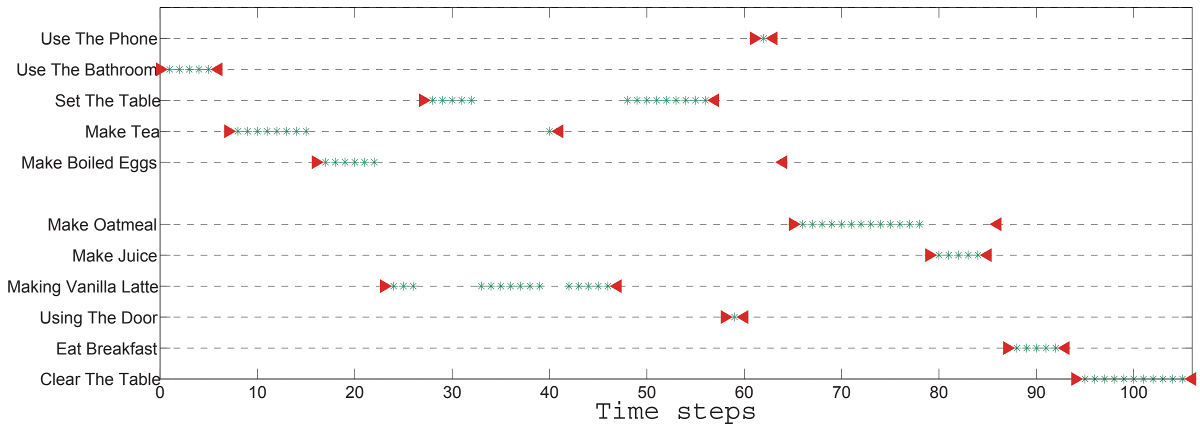

Figure 1. A sample from Patterson's assisted living dataset (Patterson [6]). The graph shows the highly interleaved nature of the activities. The red triangles mark the begin and end times of each activity. The green stars indicate when an activity is in the foreground. In this sample, for instance, up to four concurrent activities are in progress at time step 30.

Davis [4]). In previous work (Helaoui *et al.* [17]), we have started to investigate the viability of MLN for recognizing daily activities (ADL). The promising results have led to the present work attempting to address more challenging prediction tasks such as overlapping and interleaved activities. To this end, we propose a novel model with more complex temporal and inter-activity dependencies.

III. Problem Statement

Human activities of daily living (ADL) are often overlapping, alternating, and sometimes abandoned. Figure 1 depicts a typical dataset obtained from the assisted living test environment. Given observable sensor data in such a real-world assisted living environment, we are interested in inferring the current performed activity *as well as* any other activity currently in progress. The data available results uniquely from the interaction of the user with RFID tagged objects and delivers no further information about the artifacts out of the range of the wearable RFID reader. Therefore, we propose to approach the following typical activity recognition tasks:

- **The detection of the current foreground activity.** The goal is to associate each observation with the activity being actively carried out by the user (the *foreground activity*). The prediction algorithm must overcome the ambiguity due to the amount of sensor activity that several activities have in common.
- **The inference of beginning and end times.** For instance, when the activity of making a tea is interleaved with the activities of setting the table and making boiled eggs we want to determine the start and end times of each of these activities, irrespective of how many times the foreground activity changes.
- **The derivation of concurrent activities.** Unlike (Hu *el al* [1]), we define concurrent activities as being in progress simultaneously but not necessarily involving the user's interaction at the same time steps. In other words, for any two or more interleaving activities, we distinguish between *foreground* and *background* activities. At each time step, only one activity can be identified as a foreground activity involving direct user interaction. The background activities at that time step are all other activities currently in progress, that is, activities that have been started but not yet ended by the subject.

IV. Markov Logic

Markov logic combines first-order logic and undirected probabilistic graphical models (Richardson and Domingos [10]) and is one of several languages in the realm of statistical relational learning [18]. A Markov logic network (MLN) is a set of first-order formulae with weights. Intuitively, the more evidence we have that a formula is valid the higher the weight of this formula.

Markov logic can be seen as a first-order template language for log-linear models [19] with binary variables. Note that log-linear models are also known as maximum entropy models especially in the natural language processing community [20]. Markov logic has been successfully applied to problems in areas ranging from computational linguistics [21] to data integration [22]. The widespread use of Markov logic is largely due to the ease of experimentation

– objects and their relationships can be expressed in first-order formulas – and the availability of efficient learning and reasoning algorithms.

To simplify the presentation of the technical parts we do *not* include functions. In addition, we assume that all (ground) formulae of a Markov logic network are in clausal form and use the terms *formula* and *clause* interchangeably.

Syntax

A signature is a triple $S = (O, H, C)$ with O a finite set of observable predicate symbols, H a finite set of hidden predicate symbols, and C a finite set of constants. A Markov logic network (MLN) is a set of pairs $\{(F_i, w_i)\}$ with each F_i being a function-free first-order formula built using predicates from $O \cup H$ and each $w_i \in \mathbb{R}$ a real-valued weight associated with formula F_i. We can represent hard constraints using large weights.

Semantics

Let $M = (F_i, w_i)$ be a Markov logic network with signature $S = (O, H, C)$. A *grounding* of a first-order formula F is generated by substituting each occurrence of every variable in F with constants in C. Existentially quantified formulae are substituted by the disjunctions of their groundings over the finite set of constants. A formula that does not contain any variables is *ground*. A formula that consists of a single predicate is an *atom*. This definition of the semantics of Markov logic makes several assumptions such as (a) different constants refer to different objects (unique names assumption) and (b) the only objects in the domain are those representable using the constants (domain closure assumption) (Richardson and Domingos [10]). These assumptions ensure that the resulting ground Markov logic network has a finite number of nodes.

A set of ground atoms is a *possible world*. We say that a possible world W *satisfies* a formula F, and write $W \models F$, if F is true in W. Let $\mathcal{G}_F^C$ be the set of all possible groundings of formula F with respect to C. We say that W satisfies $\mathcal{G}_F^C$, and write $W \models \mathcal{G}_F^C$, if F satisfies every formula in $\mathcal{G}_F^C$. Let $\mathcal{W}$ be the set of all possible worlds with respect to S. Then, the probability of a possible world W is given by

$$p(W) = \frac{1}{Z} \exp\left(\sum_{(F_i, w_i)} \sum_{G \in \mathcal{G}_{F_i}^C :\ W \models G} w_i \right).$$

Here, Z is a normalization constant. The score s_W of a possible world W is the sum of the weights of the ground formulae implied by W

$$s_W = \sum_{(F_i, w_i)} \sum_{G \in \mathcal{G}_{F_i}^C :\ W \models G} w_i. \quad (1)$$

Table I
THE SET OF FORMULAE FOR **MODEL I**.

	Hard formulae
1	$\forall Timestep\ t\ \exists Activity\ a : currentActivity(a, t)$
2	$\forall Timestep\ t,\ Activity\ a_1,\ a_2 : [a_1 \neq a_2]$ $\Rightarrow [currentActivity(a_1, t) \Rightarrow \neg\, currentActivity(a_2, t)]$
	Soft formulae
3	$\forall Sensor\ s,\ Timestep\ t,\ Activity\ a :$ $sensor(s, t) \Rightarrow [currentActivity(a, t)]$
4	$\forall Sensor\ s_1,\ s_2,\ Timestep\ t,\ Activity\ a :$ $[sensor(s_1, t) \wedge sensor(s_2, t+1)]$ $\Rightarrow currentActivity(a, t+1)$

MAP Inference and ILP

If we want to determine the most probable state of a MLN, we need to compute the set of ground atoms of the hidden predicates that maximizes the probability given both the ground atoms of observable predicates and all ground formulae. This is an instance of MAP (maximum a-posteriori) inference in the ground Markov logic network. Let $\mathbf{O}$ be the set of all ground atoms of observable predicates and $\mathbf{H}$ be the set of all ground atoms of hidden predicates both with respect to C. We make the closed world assumption with respect to the observable predicates. Assume that we are given a set $\mathbf{O}' \subseteq \mathbf{O}$ of ground atoms of observable predicates. In order to find the most probable state of the MLN we have to compute

$$\underset{\mathbf{H}' \subseteq \mathbf{H}}{\text{argmax}} \sum_{(F_i, w_i)} \sum_{G \in \mathcal{G}_{F_i}^C :\ \mathbf{O}' \cup \mathbf{H}' \models G} w_i.$$

In this paper, every $\mathbf{H}' \subseteq \mathbf{H}$ is called a *state*. It is the set of *active* ground atoms of hidden predicates. Markov logic is by definition a declarative language, separating the formulation of a problem instance from the algorithm used for probabilistic inference. MAP inference in Markov logic networks is essentially equivalent to the (partial) weighted MAX-SAT problem and, therefore, NP-hard and APX-complete [23]. Integer linear programming (ILP) is an effective method for solving exact MAP inference in undirected graphical models (Roth and Yih [24] and Taskar [25]) and specifically in Markov logic networks (Riedel [26] and Niepert [27]). ILP is concerned with optimizing a linear objective function over a finite number of integer variables, subject to a set of linear constraints over these variables (Schrijver [28]). We omit the formal details of the ILP representation of a MAP problem and refer the reader to (Riedel [26] and Niepert [27]).

V. MARKOV LOGIC AND ACTIVITY RECOGNITION

In this section, we describe the Markov logic formulation that we used in our experimental evaluation. We distinguish between hard and soft formulae. Soft formula are created to model probabilistic dependencies between the sensors'

Table II
THE SET OF FORMULAE FOR **MODEL II**.

	Hard formulae
1	$\forall\, Timestep\, t\, \exists\ Activity\ a : currentActivity(a,t)$
2	$\forall\, Timestep\, t,\ Activity\, a_1,\ a_2 : [a_1 \neq a_2]$ $\Rightarrow [currentActivity(a_1,t) \Rightarrow \neg\, currentActivity(a_2,t)]$
3	$\forall\, Timestep\, t_1,\ t_2,\ Activity\, a :$ $startActivity(a_1,t) \Rightarrow \neg\, endActivity(a_2,t)$
4	$\forall\, Timestep\, t_1,\ t_2,\ Activity\, a :$ $[t_1 \geq t_2] \Rightarrow [currentActivity(a,t_1)$ $\Rightarrow \neg\, endActivity(a,t_2)]$
5	$\forall\, Timestep\, t_1,\ t_2,\ Activity\, a :$ $[t_1 \leq t_2] \Rightarrow [currentActivity(a,t_1)$ $\Rightarrow \neg\, startActivity(a,t_2)]$
	Soft formulae
6	$\forall\, Sensor\, s,\ Timestep\ t,\ Activity\, a :$ $sensor(s,t) \Rightarrow [currentActivity(a,t)]$
7	$\forall\, Sensor\, s_1,\ s_2,\ Timestep\ t,\ Activity\, a :$ $[sensor(s_1,t) \wedge sensor(s_2,t+1)]$ $\Rightarrow [endActivity(a,t+1) \wedge currentActivity(a,t+1)]$
8	$\forall\, Sensor\, s_1,\ s_2,\ Timestep\ t,\ Activity\, a :$ $[sensor(s_1,t+1) \wedge sensor(s_2,t)]$ $\Rightarrow [startActivity(a,t) \wedge currentActivity(a,t)]$
9	$\forall\, Timestep\ t,\ Activity\, a_1,\ a_2 : endActivity(a_1,t) \wedge$ $startActivity(a_2,t+1) \wedge currentActivity(a_2,t+1)$

events, the performed activities, and their temporal context. We employ a well-known yet simple machine learning algorithm, the voted perceptron, to learn the weights for each of the formula's groundings. In order to incorporate meaningful formulas we have to define the set of predicates in the signature. The states of sensors as well as the subject's activities are represented with binary predicates linking it to particular time steps. For instance, each grounding of the predicate $currentActivity(x,t)$ models the existence of foreground activity x at time step t. A predicate is *observable* if its truth value can be directly observed and *hidden* otherwise.

Predicting an activity at a particular time step t is equivalent to computing the MAP state of the ground Markov logic network including substituting each time variable with the constants over the finite set of the anterior time steps values. Hence, this involves the activity history up to time step $t-1$ and the sensor history up to time step t. This is due to the explicit

We evaluate two formulations within the declarative Markov logic framework. **Model I** is designed to recognize the foreground activity at each time step whereas **Model II** is more complex in that it also recognizes the beginning and end time of each activity and, therefore, all occurring activities for each time step.

Table III
BACKGROUND KNOWLEDGE AS HARD FORMULAE

1	$\forall\, Timestep\ t_1,\ t_2 :$ $[t_2 \geq t_1 + 1] \Rightarrow [currentActivity(\text{“}Clear\, The\, Table\text{”}, t_1)$ $\Rightarrow \neg currentActivity(\text{“}Eat\, Breakfast\text{”}, t_2)]$
2	$\forall\, Timestep\ t_1,\ t_2 :$ $[t_2 \geq t_1 + 1] \Rightarrow [currentActivity(\text{“}Clear\, The\, Table\text{”}, t_1)$ $\Rightarrow \neg currentActivity(\text{“}Set\, The\, Table\text{”}, t_2)]$
3	$\forall\, Timestep\ t_1,\ t_2 :$ $[t_2 \geq t_1 + 1] \Rightarrow [currentActivity(\text{“}Set\, The\, Table\text{”}, t_2)$ $\Rightarrow \neg currentActivity(\text{“}Eat\, Breakfast\text{”}, t_1)]$

Inferring foreground activity

In the formulation of **Model I** we only have to incorporate one *hidden* predicate: $currentActivity(a,t)$. For the observable data we define one predicate, $sensor(s,t)$, modelling that sensor s has been triggered at time step t. Since the sensors bear the names of the objects they tag such a predicate indicates that the user is using object s at time step t. Obviously, there is a set of constraints that has to be specified to assure logical consistency of the proposed model. These are depicted in Table I.

Note that we used *typed* predicates leading not only to a more intuitive understanding of the formulae but also smaller sized ground Markov logic networks. The *hard* formulae (1) and (2) in Table I enforce that exactly one activity can be carried out at each time step. This is a natural constraint in the ADL environment in which a user cannot interact with two RFID-tagged objects at exactly the same time step (see Section VI).

Furthermore, we include two additional *soft* formulae in the ML formulation which are depicted in Table I. Their purpose is to model dependencies between the subject's activities and the sensor activity.

Inferring foreground and background activity

In our second formulation (**Model II**), we expand **Model I** with two additional hidden predicates: $startActivity(a,t)$ and $endActivity(a,t)$. These predicates model the start and end, respectively, of activity a at time step t. Therefore, the model will now be able to infer the state of the different types of hidden predicates *jointly*, improving the overall recognition accuracy. In order to model the observable sensor activity, we use the same predicate, namely $sensor$. The set of all included formulae is depicted in Table II. Since in our use-case every activity starts and ends only once (see Section VI), we add two *hard* formulae expressing that activities cannot be finished before they start. In our future work, we intend to overcome this model limitation while applying it to more flexible datasets.

We also include a set of four *soft* formulae depicted in Table II with numbers 6-9. Similarly to **Model I**, formula (6)

Table IV
ACTIVITIES PERFORMED DURING THE MORNING DAILY ROUTINE

1	Using the bathroom	7	Preparing orange juice
2	Making soft-boiled eggs	8	Clearing the table
3	Making vanilla latte	9	Making oatmeal
4	Setting the table	10	Making tea
5	Using the door	11	Eating breakfast
6	Making a phone call		

models the dependency between the activities and the corresponding sensors. Formula (7) formalizes that the first pair of activated objects during an activity are a good indicator for its start time. Analogously, formula (8) formalizes that the last pair of activated objects during an activity is a good indicator for the activities end time. Finally, formula (9) models the temporal relationship between activities, that is, the likelihood that an activity a_1 is followed by a different activity a_2.

Incorporating Common-Sense Knowledge

The declarative nature of the ML framework allows the system designer to include background knowledge simply by adding *hard* first-order formulae. Table III, for instance, lists a set of first-order sentences expressing common-sense knowledge in the application domain. The addition of these sentences to the second model alone increased the performance of the system. The sentences specify simple temporal relationships between three activities sharing a particularly large set of common sensors. This would usually lead to several recognition problems but the experiments show that the common-sense knowledge helps to avoid most of these problems. The formulae express the following domain knowledge:

- The activity of setting the table must precede both activities of eating breakfast and clearing the table.
- The activity of eating breakfast precedes the activity of clearing the table.

Please note that with Markov logic we are able to incorporate these formulae *during* weight learning which improves the accuracy of the model.

VI. EXPERIMENTS

To evaluate our approach, we used the real data collected by Patterson $et\ al.$ [6]. The lab is outfitted with 60 RFID tags placed on different objects involved in performing a set of eleven fine-grained activities (see Table IV). The user wore two RFID gloves that triggered RFID tags within 2 inches. The data collection periods had a mean duration of 27 min per day on ten different days. The performed activities are highly interleaved in nature (see Figure 1).

The dataset comprises two subsets: "standard data" and "full data". The "standard data" is provided in form of timely ordered events relating, for each time step, the ID of

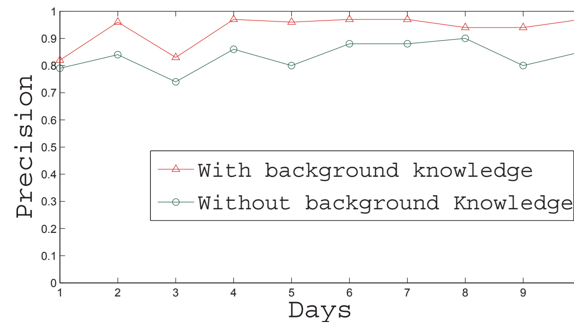

Figure 2. The average accuracy of **Model I** vs. individual days with and without common-sense background knowledge.

active sensors and the activity being carried out. Unlike the standard data, the full data provides all concurrent activities for each time step. In our work, we use the standard data for **Model I** and the full data for **Model II**. To make the learning of the weights more efficient, we omitted the input of identical successive sensor activity. We used *Markov TheBeast* [26] to convert the MLN formulations to the corresponding integer linear programming (ILP) instances.

The weights of the soft formulae were learned with a simple online learner using the perceptron rule and 15 epochs. Learning took always less than 10 seconds. For inference, we applied the mixed integer programming solver Gurobi[1] to the ILP instances. Both models were tested using ten-fold cross-validation. All experiments were conducted on a desktop PC with AMD Athlon Dual Core Processor 5400B with 2.6GHz and 1GB RAM.

As explained in Section V, the models are evaluated with three different recognition tasks:

- Predict the foreground activity for each time step with **Model I** and **Model II**.
- Infer start and end times of every activity with **Model II**.
- Derive all background activities at each event with **Model II**.

The respective results are shown in Tables V, VI, and VIII.

For the evaluation task, we apply three well-known metrics used in pattern recognition: Precision, recall and F_1 score. For each time step, we define the eleven activities as the possible labels. Correctly predicted activities are *true positives* (TP). A predicted activity a_1 at time step t that does not match the activity a_2 in the reference dataset for the same time step is counted as a *false positive* (FP). In addition, the activity a_2 counts as *false negative* (FN) since it is present in the reference dataset but missing in the prediction for the same time step. Please note that due to the fact that at each time step only one activity occurs, the number of *false positives* and *false negatives* coincide. This explains

[1] http://www.gurobi.com/

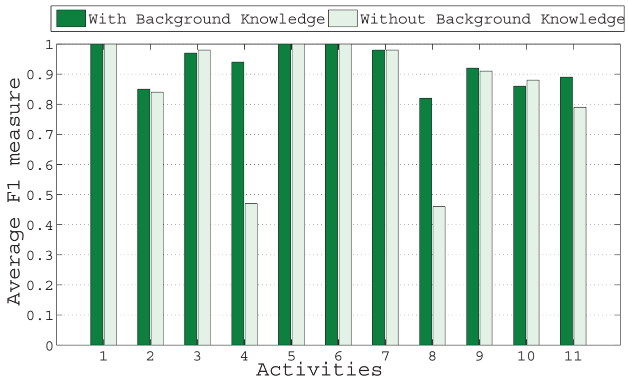

Figure 3. Average F_1 scores of **Model I** for the different activities with and without the common-sense constraints provided as background knowledge (Table III). Table IV shows the assignments of number to the individual activities. The recognition accuracy doubles for the activities "$SetTheTable$" (4) and "$ClearTheTable$" (8).

the identical precision, recall and F_1 scores.

$$\text{Precision} = \frac{TP}{(TP + FP)}$$

$$\text{Recall} = \frac{TP}{(TP + FN)}$$

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

The recognition of the foreground activities using **Model I** (Table V) outperforms state-of-the-art recognizers (Hu and Yang [13]) applied on the same dataset. Please recall that the recognition results of both models refer to two different subsets, namely the standard data (**Model I**) and the full data (**Model II**). Besides, unlike **Model I**, **Model II** has only been evaluated on the reduced version of the dataset omitting the input of identical successive sensor data. Applying the algorithm and retaining the same inferred activity until the next event, as in the case of **Model I**, might yield even better recognition results.

The set of common-sense rules depicted in Table III demonstrates the flexibility of the MLN framework to incorporate a-priori domain knowledge. Despite their simplicity, the integration of these hard constraints significantly improves the recognition accuracy. A comparison of the performance of **Model I** with and without these rules is illustrated in Figure 2. The results show the improvement of the average accuracy rising from 0.88 without the background knowledge to 0.93 with the three rather simple rules. Figure 3 gives a more detailed insight into the effect of such hard constraints. Since the activities $SetTheTable$, $EatBreakfast$, and $ClearTheTable$ share a high number of common objects, their recognition accuracy is rather low in absence of the background knowledge. However,

Table V
RESULTS FOR THE RECOGNITION OF THE CURRENT ACTIVITY USING THE STANDARD DATA FOR **MODEL I** AND THE FULL DATA FOR **MODEL II**. THE EVALUATION IS COMPUTED FROM LEAVE-ONE OUT CROSS VALIDATION.

Model	I	II
Precision	$0.93(\sigma = 0.06)$	$0.92(\sigma = 0.03)$
Recall	$0.93(\sigma = 0.06)$	$0.92(\sigma = 0.03)$
F_1	$0.93(\sigma = 0.06)$	$0.92(\sigma = 0.03)$

Table VI
RESULTS FOR THE RECOGNITION OF THE START AND END POINTS OF THE ACTIVITIES THE FULL DATA FOR **MODEL II**. THE EVALUATION IS COMPUTED FROM LEAVE-ONE OUT CROSS VALIDATION.

	Current	Start	End	Global
Precision	0.92	0.97	0.97	0.93
Recall	0.92	0.90	0.71	0.91
F_1	0.92	0.93	0.82	0.92

extending the model with the background knowledge render the accuracy almost twice as good for the activities $SetTheTable$ (4) and $ClearTheTable$ (8).

Thus, background knowledge has the potential to significantly improve the recognition accuracy. Such background knowledge could incorporate many aspects such as

- Name and/or identity of subjects in case of several subjects, their location, and their desires such sleep and hunger;
- The environment such as the temperature and the weather; and
- The time like the season of the year, day of the week, and the part of the day.

Table VI shows the viability of further detecting the start and end times of the ongoing activities. This can be of significant importance for smart environments in general and for ambient assisted living environments in particular. Indeed, a real time assessment of the activities and their durations would be possible during real-time recognition. This can improve the decision making during inference and thus improve the corresponding reactive and proactive services.

The $start$ and end predicates also enabled our system to capture the temporal relationships between the successive activities. Table VII shows some selected weights learned from the last rule in Table II. The weights coincide with the intuition that a daily "routine" preserves similar chronological order for some activities (the three first lines of the Table) and yet allows randomness for the others such as *using the phone*.

We evaluate the *concurrent* activities recognition as follows: at each sensor event the set of recognized activities is compared to all activities in progress. Each predicted activity that is absent in the reference set is counted as a *false positive*. Each activity missing in the prediction set

Table VII

SOME SELECTED WEIGHTS FOR SUCCESSIVE ACTIVITIES. THE ACTIVITIES CAN BE INTERRUPTED BY OTHERS. HIGHER WEIGHTS ARE GIVEN FOR TWO ACTIVITIES a_1 AND a_2 IF a_2 STARTS WEHN a_1 ENDS (SEE TABLE II)

Activity A_1	Following Activity A_2	Average Weight
Eat Breakfast	Clear the Table	13.1
Make Boiled Eggs	Make Oatmeal	8.6
Make Oatmeal	Make Boiled Eggs	9.8
Make Tea	Eat Breakfast	1.6
Use The Phone	Set The Table	0.0

but present in the reference set is counted as *false negative*. Correctly predicted activities are *true positives*. Finally, if an activity is not in progress and has not been predicted counts as *true negative*. To derive the predicted activities at each time step, we assume that an activity is still in progress from its predicted start point until its predicted end point. Without this assumption the overall recognition evaluation results in significantly higher values. Figure 4 and Table VIII reflect this statement which is justified by the lower recall value of the end predicate in Table VI.

VII. DISCUSSION AND FUTURE WORK

We have presented a declarative Markov logic based approach to human activity recognition. Using two different models, we have shown that Markov logic offers a simple but effective combination of statistical and relational features to accurately recognize interleaved and concurrent activities. Our models outperform state of the art algorithms applied to an established benchmark dataset. The framework offers additional features such as detecting the start and end points for each activity as well as distinguishing between foreground and background activities. One limitation of the current data is that it strictly provides sensor events from the objects in direct interaction with the user. Acquiring more information from other sensors embedded in different devices might significantly improve the recognition accuracy of background activities.

Our framework has successfully learned weights for *soft* formulae capturing temporal dependencies between individual activities. Its declarative nature allows the system developer to easily experiment with various models encoding different variables' dependencies.

Future work will be concerned with incorporating further temporal relationships and their effects on the recognition accuracy. We also plan to extend our work with automatic extraction of richer ontological background knowledge.

As mentioned above, the algorithm applies a simple weight learning algorithm, the voted perceptron. We are currently working on a novel implementation of a statistical relational learning tool tailored to activity recognition with which we hope to overcome the aforementioned limitations and apply more sophisticated weight learning methods.

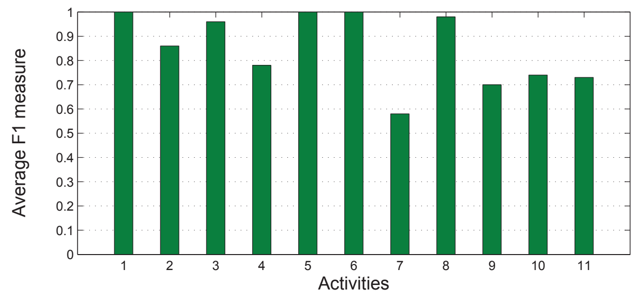

Figure 4. Average F_1 score for recognizing the foreground and all background activities at each sensor event for **Model II**. The activity legend is provided by Table IV

Table VIII

RESULTS FOR THE RECOGNITION OF CONCURRENT ACTIVITIES USING THE FULL DATA FOR **MODEL II**. THE EVALUATION IS COMPUTED FROM LEAVE-ONE OUT CROSS VALIDATION.

	Precision	Recall	F_1
Micro-average	0.68	0.94	0.78
Macro-average	0.83	0.95	0.88

Finally, we intended to evaluate our statistical relational models with other datasets including more heterogeneous sensor data and context specific information.

REFERENCES

[1] D. Hao Hu, S. J. Pan, V. W. Zheng, N. N. Liu, and Q. Yang, "Real world activity recognition with multiple goals," in *UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing*. New York, NY, USA: ACM, 2008, pp. 30–39.

[2] M. Cirillo, F. Lazellotto, F. Pecora, and A. Saffiotti, "Monitoring domestic activities with temporal constraints and components," in *Proc of the 5th Int Conf on Intelligent Environments (IE09)*, Barcelona, ES, 2009, pp. 117–124.

[3] R. Biswas, S. Thrun, and K. Fujimura, "Recognizing activities with multiple cues," in *HUMO'07: Workshop on Human Motion*, 2007, pp. 255–270.

[4] S. D. Tran and L. S. Davis, "Event modeling and recognition using markov logic networks," in *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 610–623.

[5] M. Buettner, R. Prasad, M. Philipose, and D. Wetherall, "Recognizing daily activities with rfid-based sensors," in *Ubicomp '09: Proceedings of the 11th international conference on Ubiquitous computing*. New York, NY, USA: ACM, 2009, pp. 51–60.

[6] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose, "Fine-grained activity recognition by aggregating abstract object usage," in *ISWC '05: Proceedings of the Ninth IEEE International Symposium on Wearable Computers*. Washington, DC, USA: IEEE Computer Society, 2005.

[7] M. Stikic and B. Schiele, "Activity recognition from sparsely labeled data using multi-instance learning," in *LoCA*. Springer, 2009, pp. 156–173.

[8] T. Gu, S. Chen, X. Tao, and J. Lu, "An unsupervised approach to activity recognition and segmentation based on object-use fingerprints," *Data Knowledge Engineering*, vol. 69, no. 6, pp. 533–544, 2010.

[9] H. Jeon, T. Kim, and J. Choi, "Ontology-based user intention recognition for proactive planning of intelligent robot behavior," in *MUE '08: Proceedings of the 2008 International Conference on Multimedia and Ubiquitous Engineering*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 244–248.

[10] M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning*, vol. 62, pp. 107–136, 2006.

[11] R. Helaoui, "Towards a proactive system based on activity recognition," in *PerCom '10: Proceedings of the Eighth Annual IEEE International Conference on Pervasive Computing and Communications*, 2010.

[12] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *IEEE Pervasive Computing*, vol. 9, pp. 48–53, 2010.

[13] D. H. Hu and Q. Yang, "Cigar: Concurrent and interleaving goal and activity recognition," in *AAAI'08:Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, 2008, pp. 1363–1368.

[14] A. McCallum and C. Sutton, "An introduction to conditional random fields for relational learning," in *Introduction to Statistical Relational Learning*, L. Getoor and B. Taskar, Eds. MIT Press, 2006.

[15] J. Modayil, T. Bai, and H. Kautz, "Improving the recognition of interleaved activities," in *UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing*, New York, NY, USA, 2008, pp. 40–43.

[16] D. Riboni and C. Bettini, "Context-aware activity recognition through a combination of ontological and statistical reasoning," in *UIC '09: Proceedings of the 6th International Conference on Ubiquitous Intelligence and Computing*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 39–53.

[17] R. Helaoui, M. Niepert, and H. Stuckenschmidt, "A statistical-relational activity recognition framework for ambient assisted living systems," in *Ambient Intelligence and Future Trends-International Symposium on Ambient Intelligence*, ser. Advances in Soft Computing, vol. 72/2010, 2010, pp. 247–254.

[18] L. Getoor and B. Taskar, *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[19] D. Koller and N. Friedman, *Probabilistic Graphical Models*. MIT Press, 2009.

[20] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *Proceedings of the 40th Meeting of the ACL*, 2002.

[21] I. Meza-Ruiz and S. Riedel, "Multilingual semantic role labelling with markov logic," in *Proceedings of the Conference on Computational Natural Language Learning*, 2009, pp. 85–90.

[22] M. Niepert, C. Meilicke, and H. Stuckenschmidt, "A Probabilistic-Logical Framework for Ontology Matching," in *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 2010.

[23] N. Creignou, S. Khanna, and M. Sudan, *Complexity classifications of boolean constraint satisfaction problems*. SIAM, 2001.

[24] D. Roth and W. Yih, "Integer linear programming inference for conditional random fields," in *Proceedings of the International Conference on Machine Learning*, 2005, pp. 736–743.

[25] B. Taskar, "Learning structured prediction models: A large-margin approach," Ph.D. dissertation, Stanford University, 2004.

[26] S. Riedel, "Improving the accuracy and efficiency of map inference for markov logic," in *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2008, pp. 468–475.

[27] M. Niepert, "A delayed column generation strategy for exact k-bounded map inference in markov logic networks," in *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, 2010.

[28] A. Schrijver, *Theory of Linear and Integer Programming*. Wiley & Sons, 1998.