Unsupervised Recognition of Interleaved Activities of Daily Living through Ontological and Probabilistic Reasoning

Daniele Riboni
Univ. of Cagliari, Italy
riboni@unica.it

Timo Sztyler
Univ. of Mannheim, Germany
timo@informatik.uni-mannheim.de

Gabriele Civitarese
Univ. of Milano, Italy
gabriele.civitarese@unimi.it

Heiner Stuckenschmidt
Univ. of Mannheim, Germany
heiner@informatik.uni-mannheim.de

ABSTRACT

Recognition of activities of daily living (ADLs) is an enabling technology for several ubiquitous computing applications. In this field, most activity recognition systems rely on supervised learning methods to extract activity models from labeled datasets. An inherent problem of that approach consists in the acquisition of comprehensive activity datasets, which is expensive and may violate individuals' privacy. The problem is particularly challenging when focusing on complex ADLs, which are characterized by large intra- and inter-personal variability of execution. In this paper, we propose an unsupervised method to recognize complex ADLs exploiting the semantics of activities, context data, and sensing devices. Through ontological reasoning, we derive semantic correlations among activities and sensor events. By matching observed sensor events with semantic correlations, a statistical reasoner formulates initial hypotheses about the occurred activities. Those hypotheses are refined through probabilistic reasoning, exploiting semantic constraints derived from the ontology. Extensive experiments with real-world datasets show that the accuracy of our unsupervised method is comparable to the one of state of the art supervised approaches.

ACM Classification Keywords

I.2.4 Knowledge Representation Formalisms and Methods

Author Keywords

Activity recognition; Ontological reasoning; Probabilistic reasoning

INTRODUCTION

Knowledge about the activities carried out by individuals is a requirement for several ubiquitous computing applications [9]. Indeed, domains of activity-aware computing range from smart homes and e-health, to gaming, smart manufacturing, pervasive advertising, and smart cities. In particular, the rapid growing of the population age in industrialized societies advocates activity-aware systems to support active and healthy ageing. The goal of such systems is to early recognize health issues and prolong independent life [10, 24]. Hence, many ubiquitous healthcare systems have been proposed, which monitor activities of daily living (ADLs) at home through unobtrusive sensors and artificial intelligence methods.

Currently, most activity recognition systems rely on supervised learning applied to datasets of activities and sensor data [2, 3]. Supervised learning proves to be effective in recognizing activities characterized by specific postures or motions, such as for physical activities. However, its applicability to complex ADLs (e.g., cooking, cleaning, and dressing) is questionable. On the one side, the way in which individuals perform ADLs strongly depends on current context conditions. Hence, a large dataset of ADLs should be acquired to capture most execution patterns in different situations. On the other side, activity execution patterns are strongly coupled to the individual's characteristics and home environment, and the portability of activity datasets is an open issue [8]. As a consequence, ideally one extensive ADLs dataset should be acquired from each monitored individual. Unfortunately, acquiring ADLs datasets is very expensive in terms of annotation costs [4, 25]. Besides, activity annotation by an external observer, by means of cameras or direct observation, violates the user's privacy.

To overcome that problem, other works relied on knowledge-based activity models, manually specified through logic languages and ontologies. Those models are matched with acquired sensor data to recognize the activities [18, 33, 6]. However, the main shortcoming of that approach relies in the rigidity of specifications. For instance, complex ADLs are often specified through temporal sequences of simpler actions [14]. Nevertheless, it is unfeasible to enumerate all the possible sequences of actions describing a complex ADL.

In this work, we propose a method to overcome the limitations of both approaches. First, our method is unsupervised: we do

UbiComp '16, September 12-16, 2016, Heidelberg, Germany
©2016 ACM. ISBN 978-1-4503-4461-6/16/09...$15.00
DOI: http://dx.doi.org/10.1145/2971648.2971691

not need to acquire expensive activity datasets. Second, our activity model is based on general semantic relations among activities and smart-home infrastructure; hence, we can seamlessly reuse our model with different individuals and in different environments. Specifically, we defined an OWL 2 [12] ontology to formally model the smart home environment and the semantics of activities. We rely on ontological reasoning to derive necessary conditions about the sensor events that must occur during the execution of a specific activity in the current environment. This also enables to extract *semantic correlations* among fired sensor events and executed ADLs. Based on the semantic correlations, a statistical algorithm pre-processes sensor events to identify *candidate* activity instances, i.e., initial hypotheses about the start and end time of occurred activities. Finally, we translate our ontological model in a Markov Logic Network (MLN) [30], and perform probabilistic reasoning to refine candidate activity instances and check their consistency. Our MLN model is carefully crafted to support the recognition of interleaved activities.

We performed extensive experiments with real-world datasets of ADLs performed by 22 individuals in two different smart home environments. Results show that, even using a smaller number of sensors, the performance of our unsupervised method is comparable to the one of existing methods that rely on labeled activity datasets.

The main contributions of our work are the following:

- we propose an unsupervised method that overcomes the main drawbacks of supervised and specification-based approaches;
- we rely on general semantic properties that can be seamlessly reused with different individuals and environments;
- we recognize interleaved activities, while many other works are restricted to sequential ones.

The paper is structured as follows: After presenting related work and preliminary notions, we introduce our model and system. Then, we describe our ontological and probabilistic activity recognition methods in detail, followed by the experimental evaluation. We conclude with final remarks and future research directions.

RELATED WORK

Activity recognition methods in ubiquitous computing can be broadly classified in two categories: learning-based methods and specification-based methods [35].

Learning-based methods rely on a training set of sensor data, labeled with executed activities, and supervised learning algorithms to build the activities' model. Physical activity recognition systems are mainly based on data acquired from body-worn accelerometers [2, 3, 32]. The same approach is extended with the use of environmental data acquired from other sensors (e.g., microphones) to recognize ADLs [17, 19]. Observations regarding the user's surrounding environment (in particular, objects' use), possibly coupled with body-worn sensor data, are the basis of other activity recognition systems [13, 29]. However, since training data is hard to acquire in realistic environments, systems relying on supervised learning are prone to serious scalability issues the more activities and the more context data are considered. Moreover, datasets of complex ADLs are strongly coupled to the environment in which they are acquired (i.e., the home environment and the sensors setup), and to the mode of execution of the specific individual. Hence, the portability of activity datasets in different environments is an open issue [8]. In this work, we propose a method to recognize complex ADLs through semantic reasoning, even without the use of training data. However, when training data is available, we can exploit it to mine low-level dependencies between sensor events and performed activities. Those relationships are used by our probabilistic ontological reasoner to identify occurred activities. The combination of specification-based and probabilistic approaches has also been investigated in other fields of Artificial Intelligence [11]. However, our method supports the recognition of interleaved activities, while most existing techniques are restricted to sequential ones. We target the recognition of interleaved activities explicitly by considering this aspect in our *MLN* model, where sensor events can be assigned to overlapping activity instances.

Unsupervised learning algorithms build activity models relying on a training set of unlabeled sensor data. Some methods analyze textual descriptions of activities mined from the Web in order to obtain correlations among used objects and activities [23, 34]. In our work, we mine not only correlations, but also necessary conditions about sensor events that must be observed during the activity execution. Moreover, we derive correlations and necessary conditions considering the actual environment where activities are executed, while methods based on Web mining derive generic correlations. An unsupervised method that is close to our approach has been proposed by Ye et al. [37], where ontologies are used to derive semantic similarity between sensor events. This similarity is used to segment sensor data, obtaining sequential activities' patterns used to train a clustering model. With respect to that work, our method is totally independent from the data and it also considers interleaved activities.

Specification-based methods rely on knowledge-based definitions of the characteristics and semantics of complex activities. These are matched with available sensor data to recognize the current activity. Those definitions are usually expressed through logical axioms, rules, or description logics [18, 27, 33]. Background knowledge of ADLs has been used to create activity models, used to recognize ADLs based on the similarity of sequences of sensor events to the general models [15]. Ontological reasoning has also been proposed to perform dynamic segmentation of sensor data [20, 22, 36] or to refine the output of supervised learning methods [26]. Defeasible reasoning has been adopted to enhance existing sequential activity recognition systems by detecting interleaved activities and handling inconsistent or conflicting information [21]. Further, probabilistic description logics have been used to recognize ADLs considering the variability of activity execution [14]. However, those works rely on rigid assumptions about the simpler constituents of activities. Hence, while the specification-based approach is effective for activities characterized by a few typical execution patterns, it is hardly scalable to the comprehensive specification of complex ADLs in dif-

ferent contexts. On the contrary, in this work we rely on general semantic relations among activities and smart-home infrastructure, which are fine-tuned to the current context.

PRELIMINARIES

In this section, we introduce preliminary notions about description logics and Markov Logic Networks.

Description logics and formal ontologies

In computer science, description logics (DLs) [1] have emerged as the state-of-the-art formalism to represent *ontologies*. These enable to formally define concepts of a domain of interest, their properties, and the relationships among concepts. In this work, we use an ontology to formally define the semantics of activities, sensor events, and context data. Moreover, DLs support ontological reasoning, which allows to verify the consistency of the knowledge base, and to infer additional information from existing facts. The formalism of choice is typically OWL 2 [12]. A knowledge engineer can model the domain of interest by means of classes, individuals, properties of individuals, and relationships among individuals. Several operators can be used to declare complex definitions based on simpler ones, including operators for conjunction, disjunction, negation, and universal and existential quantification. For instance, the activity PREPARINGHOTMEAL can be defined based on the definitions of PREPARINGMEAL and PREPARINGCOLDMEAL:

$$\text{PREPARINGHOTMEAL} \equiv \text{PREPARINGMEAL} \sqcap \neg \text{PREPARINGCOLDMEAL}$$

In this work, we also exploit the following operators:

- *Qualified cardinality restriction* restricts the class membership to those instances that are in a given relation with a minimum or maximum number of other individuals of a given class. For instance, the following axiom states that `PreparingHotMeal` requires the use of at least one instrument to cook food:

$$\text{PREPARINGHOTMEAL} \sqsubseteq \text{ACTIVITY} \sqcap \geq 1\, \text{REQUIRESUSAGEOF.COOKINGINSTRUMENT}$$

- *Composition of properties.* OWL 2 supports a restricted form of property composition $\circ$. For instance, the following axiom states that if a person is in a given apartment, and she is executing a given activity, then that activity is executed in that apartment:

$$\text{EXECUTESACT}^{-} \circ \text{ISINLOCATION} \sqsubseteq \text{ACTISEXECUTEDINLOCATION}$$

Note that $\text{EXECUTESACTIVITY}^{-}$ denotes the inverse of EXECUTESACTIVITY.

Formally, a DLs knowledge base is composed by a pair $\langle \mathscr{T}, \mathscr{A} \rangle$. The *TBox* $\mathscr{T}$ constitutes the terminological part of the knowledge base. The TBox is composed of a set of axioms $C \sqsubseteq D$ or $P \sqsubseteq R$ (*inclusions*) and $C \equiv D$ or $P \equiv R$ (*equality*), where C and D are classes, and P and R are object properties. An axiom $C \sqsubseteq D$ is satisfied by an interpretation $\mathscr{I}$ when $C^{\mathscr{I}} \subseteq D^{\mathscr{I}}$. An interpretation $\mathscr{I}$ satisfies a TBox $\mathscr{T}$ when $\mathscr{I}$ satisfies all the axioms of $\mathscr{T}$.

The *ABox* $\mathscr{A}$ is the assertional part of the knowledge base. The ABox is composed of a set of axioms of the form $x : C$ and $\langle x, y \rangle : R$, where x and y are individuals, C is a class, and R is an object property. For instance, "MARY : ELDERLYPERSON" denotes that Mary is an elderly person and "$\langle$ MARY, APARTMENT23 $\rangle$: LIVESIN" represents that Mary lives in Apartment23. Axioms $x : C$ and $\langle x, y \rangle : P$ are satisfied by an interpretation $\mathscr{I}$ when $x^{\mathscr{I}} \in C^{\mathscr{I}}$ and $\langle x^{\mathscr{I}}, y^{\mathscr{I}} \rangle \in P^{\mathscr{I}}$, respectively. An interpretation $\mathscr{I}$ satisfies an ABox $\mathscr{A}$ when $\mathscr{I}$ satisfies all the axioms of $\mathscr{A}$. An interpretation $\mathscr{I}$ that satisfies both the TBox $\mathscr{T}$ and the ABox $\mathscr{A}$ is called a *model* of $\langle \mathscr{T}, \mathscr{A} \rangle$. DLs support several reasoning tasks. In particular, we rely on the following ones:

- *Satisfiability:* a class C is satisfiable with respect to a TBox $\mathscr{T}$ if there exists a model $\mathscr{I}$ of $\mathscr{T}$ such that $C^{\mathscr{I}}$ is non empty. We execute this reasoning task to check the consistency of our ontological model.
- *Property fillers retrieval:* retrieving all the instances in $\mathscr{A}$ that are related to a given individual with respect to a given property. We execute this reasoning task to derive semantic correlations among activities and sensor events.

Markov Logic with numerical Constraints

In addition to purely logical or probabilistic approaches, a Markov Logic Network provides many benefits and allows to handle uncertainty, imperfection, and contradictory knowledge. These characteristics make it an appealing tool to reason with sensor data and ADLs. Technically, a Markov Logic Network (MLN) $\mathscr{M}$ is a finite set of pairs $(F_i, w_i), 1 \leq i \leq n$, where each F_i is an axiom in function-free first-order logic and $w_i \in \mathbb{R}$ [30]. Together with a finite set of constants $C = \{c_1, ..., c_n\}$ it defines the *ground* MLN $\mathscr{M}_{\mathscr{C}}$, i.e., the MLN in which axioms do not contain any free variables. This comprises one binary variable for each grounding of F_i with weight w_i. Hence, a MLN defines a log-linear probability distribution over Herbrand interpretations (possible worlds)

$$P(\mathbf{x}) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(\mathbf{x}) \right) \tag{1}$$

where $n_i(\mathbf{x})$ is the number of satisfied groundings of F_i in the possible world $\mathbf{x}$ and Z is a normalization constant.

In a previous work, we extended MLN with numerical constraints resulting in a formalism denoted $\mathsf{MLN_{NC}}$ [16, 5]. In this work, we use this extension to reason on the temporal domain of activities and sensor events. The constraints are predicates of the form $\theta \bowtie \psi$, where θ and ψ denote variables, numerical constants, or algebraic expressions (that might contain elementary operators). In this context, the binary operator $\bowtie$ returns a truth value under a particular grounding.

DEFINITION ($\mathsf{MLN_{NC}}$). *A numerical constraint* NC *is composed of numerical constants (e.g., elements of* $\mathbb{N}$, $\mathbb{I}$*), variables, elementary operators or functions* $(+, *, -, \div, \%, \sqrt{})$*, standard relations* $(>, <, =, \neq, \geq, \leq)$*, and Boolean operators* $(\wedge, \vee)$*. An* $\mathsf{MLN_{NC}}$ *is a set of pairs* (FC_i, w_i) *where* FC_i *is a formula in first-order logic that may contain a* NC *and* w_i *is a real number representing the weight of* FC_i.

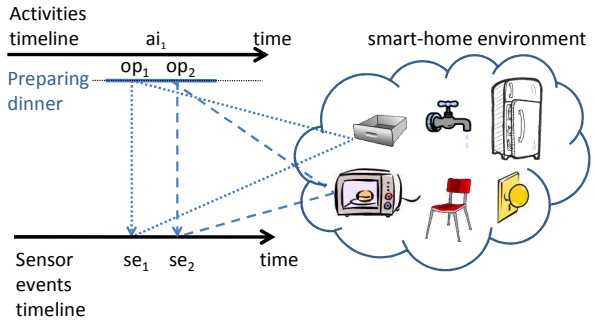

Figure 1. The time-lines illustrate the relations between an activity instance, performed operations, and the involved sensor events. Sensors detect interaction with items and furniture, or presence in a certain area.

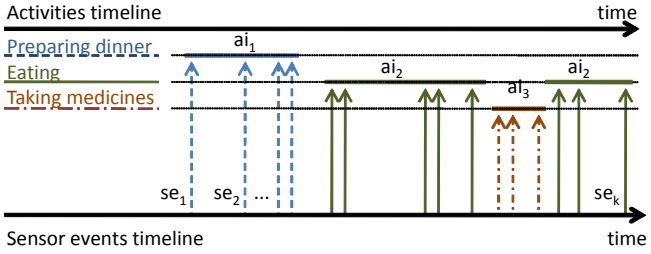

Figure 2. Reconstruction of the activity instances generating a set of sensor events. The individual sensor events, their relations, and dependencies indicate by which activity they were generated.

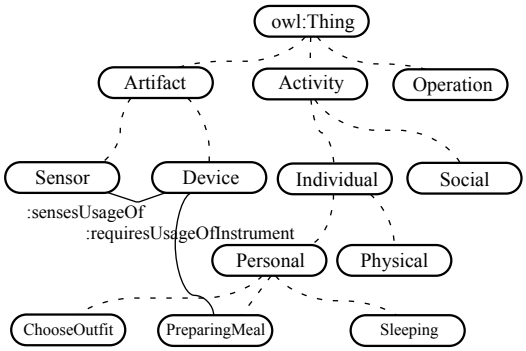

Figure 3. Excerpt of our ontology. The dashed lines represent a *subClassOf* relation where the upper is the parent of the lower class. In addition, the individual classes have relations that describe dependencies.

EXAMPLE 1. *Using* $\mathsf{MLN}_{\mathsf{NC}}$ *it is possible to represent the axiom: two events "turning on the oven" cannot belong to the same instance of meal preparation if their temporal distance is more than two hours:*
$\{\forall\, se_1, se_2, ai_1, ai_2, t_1, t_2 : event(se_1, 'oven', t_1) \wedge event(se_2, 'oven', t_2) \wedge occursIn(se_1, ai_1) \wedge occursIn(se_2, ai_2) \wedge \mathsf{NC}(t_1, t_2) \Rightarrow ai_1 \neq ai_2, \mathsf{NC}(t_1, t_2) = |t_1 - t_2| > 120\}$.

Maximum a posteriori (MAP) inference is the task of finding the most probable world given some observations also referred to as evidence. Given the observed variables $E = e$, the MAP problem aims to find an assignment of all non-evidence (hidden) variables $X = x$ such that $\mathbf{I} = \underset{x}{\operatorname{argmax}}\, P(X = x \mid E = e)$.

Based on the MLN of sensor events and semantic constraints, we apply MAP inference to derive the most probable activities.

MODEL AND SYSTEM OVERVIEW

We assume a smart home instrumented with sensors to detect interactions with items and furniture, context conditions (e.g., temperature), and presence in certain locations. We denote by *activity class* an abstract activity (e.g., cooking and cleaning), and by *activity instance* the actual occurrence of an activity of a given class during a certain time period. In this context, Figure 1 illustrates the relation between recorded sensor events and an activity instance. Hence, during the execution of activity instance ai_1 (*preparing dinner*), the subject executes the operations op_1 (opening the silverware drawer) and op_2 (turning on the microwave oven). Supposing that sensors are available to detect these operations, op_1 and op_2 generate two sensor events se_1 and se_2, whose timestamp corresponds to the time of the respective operation.

Based on the observation of a set of timestamped sensor events, the goal of the activity recognition system is to reconstruct the most probable activity instances that generated them. As shown in Figure 2, we achieve this goal by assigning each event se_i to the activity instance that most probably generated it. This approach allows us to recognize interleaved activities, as it is the case for ai_2 and ai_3 (the subject temporarily interrupts her meal to take medicines). In the following, we outline our model, architecture, and the individual components.

Ontological model

We have defined the semantics of activities and operations in an OWL 2 ontology. In the following, we consider $\mathbf{A} = \{ac_1, ac_2, \ldots, ac_k\}$ as the set of activity classes. Further, an instance ai_i of an activity class $ac_j \in \mathbf{A}$ represents the occurrence of ac_j during a given timespan. The activity instance is associated to the operations executed to perform it, where the start and end time of instances of different activities can overlap. This means that the head and the tail of a sub-sequence of sensor events se_i could be assigned to a specific instance a_j, while the sensor events that occur between them may be assigned to another instance a_k (see Figure 2). This enables us to explicitly handle interleaved activities.

Figure 3 illustrates an excerpt of our ontology, which models a complete home environment. In addition, it also covers axioms for each activity class that describe dependencies and conditions. In particular, we express necessary conditions for a set of operations to be generated by an instance of that class, according to the activity semantics. For example, the operations generated by an instance of *preparing hot meal* must include an operation *using a cooking instrument*. The ontology also models sensors and the operation that they detect; e.g., a power sensor attached to the electric stove detects the operation *turning on the stove*. In turn, this operation is a subclass of *using a cooking instrument*. The ontology carefully describes these kind of relations and, through ontological reasoning, we can derive constraints like the following: "since the stove is the only cooking instrument in the home, and a sensor is available that detects the usage of the stove, then each instance of *preparing hot meal* executed in the home must necessarily generate an event from that sensor".

Besides, other necessary conditions regard time and location. This includes constraints on the duration of the activity in-

stance, and dependencies between activity and location. As explained in the next section, ontological reasoning is also used to infer probabilistic dependencies among sensor event types and classes of executed activities; we denote them as *semantic correlations*. Our ontology is publicly available[1].

Architecture

Figure 4 shows an overview of our system. The smart-home monitoring system collects raw events data from the sensor network, including environmental, presence, and contact sensors. The SEMANTIC INTEGRATION LAYER applies simple pre-processing rules to detect operations from raw sensor events. For example, if at time t the fridge door sensor produces the raw event *open*, then the operation at t is *opening the fridge*. We denote $\mathbf{E}$ as the set of pre-processed event types that correspond to the set of monitored operations (e.g., $\mathbf{E} = \{$ *opening_the_fridge*, *closing_the_fridge* $\}$). In addition, $\mathbf{T}$ describes the set of all possible event timestamps. A temporally-ordered set of events is represented as follows:

$$\langle Event(se_1, et_1, t_1), \ldots, Event(se_k, et_k, t_k) \rangle,$$

where $Event(se_i, et_i, t_i)$ indicates that se_i is an instance of the event type $et_i \in \mathbf{E}$ occurred at timestamp $t_i \in \mathbf{T}$. A unique timestamp is assigned to each event.

The SEMANTIC CORRELATION REASONER performs ontological reasoning to derive semantic correlations among event types and activity classes; e.g., "the event type *UseStove* is strongly related to *PreparingHotMeal* and unrelated to *PreparingColdMeal*". Those correlations are used by the module for STATISTICAL ANALYSIS OF EVENTS to identify *candidate* activity instances, which are then refined by the MLN_{NC} reasoner. In particular, the events as well as the candidate activity instances are used to populate the assertional part of the MLN_{NC} knowledge base. The ontological model of considered activities and events is translated into the MLN_{NC} model. Periodically (e.g., at the end of each day), MAP inference is performed to assign each event to the candidate activity instance that most probably generated it, according to semantic correlations and ontological constraints. Finally, the output of MAP inference is post-processed to detect the exact start and end time of occurred activity instances.

ONTOLOGICAL REASONING

In the following, we introduce a simple running example to illustrate our approach.

EXAMPLE 2. *Suppose to monitor three activities in a smart home: preparing hot meal, preparing cold meal, and preparing tea. The home contains: one silverware drawer, one stove, and one freezer, each equipped with a sensor to detect its usage. No training set of activities is available. How can we exploit semantic reasoning to recognize the activities?*

In the following of this section, we explain how we answer the above question.

Semantic correlation reasoner

The specific objective of this reasoner is to compute the degree of correlation among sensor events and the activities

[1] http://sensor.informatik.uni-mannheim.de/#results2016unsupervised

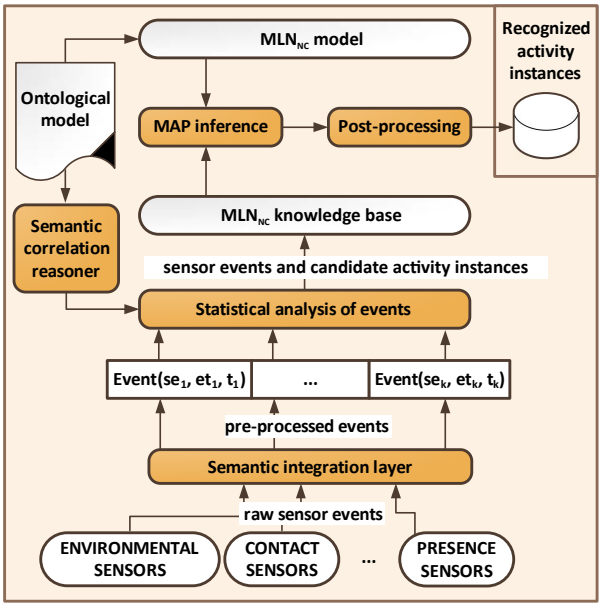

Figure 4. System overview. The *statistical analysis* layer combines the information received from the sensors and the ontological model to build a *knowledge base*. *MAP inference* enables to derive the most probable world from this knowledge base considering the MLN_{NC} *model*. This results in the recognition of the actual activity instances.

performed in the home. As illustrated in the axioms below, in our ontology, artifacts are organized in a hierarchy. The class STOVE is a sub-class of cooking instruments, used in the apartment to prepare hot meal or tea, where FREEZER is a DEVICE used to prepare hot or cold meal. SILVERWARE-DRAWER belongs to FOODPREPFURNITURE and is used for the three activities. The instance {APT} represents the current apartment. For clarity, we represent the name of ontological instances within curly brackets.

$$\text{STOVE} \sqsubseteq \text{COOKINGINSTRUMENT} \sqcap \Big(\exists \text{USEDFOR}.((\text{PREPHOTMEAL} \sqcup \text{PREPTEA}) \sqcap (\exists \text{OCCURSIN}.\{\text{APT}\}))\Big).$$

$$\text{FREEZER} \sqsubseteq \text{DEVICE} \sqcap \Big(\exists \text{USEDFOR}.((\text{PREPHOTMEAL} \sqcup \text{PREPCOLDMEAL}) \sqcap (\exists \text{OCCURSIN}.\{\text{APT}\}))\Big).$$

$$\text{SILVERWAREDRAWER} \sqsubseteq \text{FOODPREPFURNITURE}.$$

$$\text{FOODPREPFURNITURE} \sqsubseteq \text{FURNITURE} \sqcap \Big(\exists \text{USEDFOR}.((\text{PREPTEA} \sqcup \text{PREPCOLDMEAL} \sqcup \text{PREPHOTMEAL}) \sqcap (\exists \text{OCCURSIN}.\{\text{APT}\}))\Big).$$

Based on the smart home setup, we instantiate the ontology with the sensors and artifacts in the apartment, and we specify which activities we want to monitor.

EXAMPLE 3. *The activities that we want to monitor are* {AC_PREP_COLD_MEAL}, {AC_PREP_HOT_MEAL} *and* {AC_PREP_TEA}. *They are instances representing the generic occurrences of* PREPCOLDMEAL, PREPHOTMEAL, *and* PREPTEA, *respectively. Lines 5 and 6 state that at most one*

*instance of each activity type can be monitored at a time. Further, lines 7 and 8 represent that the {*APT*} contains exactly one cooking instrument, one silverware drawer, and a freezer:*

$$\{\text{APT}\} = \text{APARTMENT} \quad (1)$$
$$\sqcap\big(\exists\text{MONITACT.}(\{\text{AC\_PREP\_COLD\_MEAL}\})\big) \quad (2)$$
$$\sqcap\big(\exists\text{MONITACT.}(\{\text{AC\_PREP\_HOT\_MEAL}\})\big) \quad (3)$$
$$\sqcap\big(\exists\text{MONITACT.}(\{\text{AC\_PREP\_TEA}\})\big) \quad (4)$$
$$\sqcap(\leq 1\text{MONITACT.PREPCOLDMEAL}) \quad (5)$$
$$\sqcap(\leq 1\text{MONITACT.PREPHOTMEAL})\sqcap(\leq 1\text{MONITACT.PREPTEA}) \quad (6)$$
$$\sqcap(= 1(\text{ISIN})^{-}.\text{COOKINGINSTRUMENT}) \quad (7)$$
$$\sqcap(= 1(\text{ISIN})^{-}.\text{SILVERWAREDRAWER})\sqcap(= 1(\text{ISIN})^{-}.\text{FREEZER}). \quad (8)$$

Subsequently, we introduce an instance in the ontology for each artifact in the apartment:

$$\{\text{STOVE}\} \equiv \text{STOVE}\sqcap\exists\text{ISIN.}\{\text{APT}\}.$$
$$\{\text{FREEZER}\} \equiv \text{FREEZER}\sqcap\exists\text{ISIN.}\{\text{APT}\}.$$
$$\{\text{SILVERWARE\_DRAWER}\} \equiv \text{SILVERWAREDRAWER}\sqcap\exists\text{ISIN.}\{\text{APT}\}.$$

We also instantiate each sensor that occurs in our apartment:

$$\{\text{S\_STOVE}\} \equiv \text{POWERSENSOR}\sqcap(\exists\text{SENSESUSAGEOF.}\{\text{STOVE}\})$$
$$\sqcap(\exists\text{PRODUCESEVENT.}\{\text{ET\_STOVE}\}).$$

$$\{\text{S\_SILVERWARE\_DRAWER}\} \equiv \text{CONTACTSENSOR}$$
$$\sqcap(\exists\text{SENSESUSAGEOF.}\{\text{SILVERWARE\_DRAWER}\})$$
$$\sqcap(\exists\text{PRODUCESEVENT.}\{\text{ET\_SILVERWARE\_DRAWER}\}).$$

$$\{\text{S\_FREEZER}\} \equiv \text{CONTACTSENSOR}$$
$$\sqcap(\exists\text{SENSESUSAGEOF.}\{\text{FREEZER}\})$$
$$\sqcap(\exists\text{PRODUCESEVENT.}\{\text{ET\_FREEZER}\}).$$

According to the introduced axioms, {S_STOVE} is an instance of POWERSENSOR which senses the usage of {STOVE} and produces a generic event of type {ET_STOVE}. Similarly, the last two axioms define sensors and events for the silverware drawer and the freezer, respectively.

We exploit the property composition operator to infer the semantic correlations between sensor events and activity types. In particular, we use the following axiom, which states that: "if an event of type *et* is produced by a sensor that detects the usage of an artifact possibly used for an activity of class *ac*, then *et* is a *predictive sensor event type* for *ac*":

$$\text{PRODUCESEVENT}^{-}\circ\text{SENSESUSAGEOF}\circ$$
$$\text{USEDFOR}\rightarrow\text{PREDICTIVESENSOREVENTFOR}$$

Then, we perform ontological reasoning to infer the fillers of property PREDICTIVESENSOREVENTFOR, and use them to compute semantic correlations.

EXAMPLE 4. *Considering all of the introduced axioms, the OWL 2 reasoner infers that:*

- {ET_STOVE} *is a predictive sensor event type for* {AC_PREPARE_HOT_MEAL} *and* {AC_PREP_TEA}.
- {ET_SILVERWARE_DRAWER} *is a predictive sensor event type for* {AC_PREP_HOT_MEAL},{AC_PREP_COLD_MEAL} *and* {AC_PREP_TEA}.
- {ET_FREEZER} *is a predictive sensor event type for* {AC_PREP_HOT_MEAL} *and* {AC_PREP_COLD_ MEAL}.

We represent semantic correlations using a *prior probability matrix* (*PPM*). The rows correspond to the activity classes, where the columns to the sensor event types. Hence, $PPM(ac, et)$ stores the probability of an event of type *et* being generated by an activity of class *ac*. If a given sensor event type is predictive of a single activity class, the value of the corresponding entry is 1; if it is predictive of multiple activity classes, the value is uniformly distributed among them. The prior probability matrix resulting from our running example is shown in Table 1. The PPM is given as input to the STATISTICAL ANALYSIS OF EVENTS module.

	{et_stove}	**{et_silverware_drawer}**	**{et_freezer}**
{ac_prep_hot_meal}	0.5	$0.\overline{33}$	0.5
{ac_prep_cold_meal}	0.0	$0.\overline{33}$	0.5
{ac_prep_tea}	0.5	$0.\overline{33}$	0.0

Table 1. Prior probability matrix of our running example.

Deriving necessary sensor observations

Our ontology includes a property REQUIRESUSAGEOFARTIFACT, which associates artifacts in the apartment with activities for which they are necessary.

EXAMPLE 5. *Continuing our running example, the axiom below defines* PREPHOTMEAL *as a subclass of* PREPAREMEAL *that requires the usage of a cooking instrument:*

$$\text{PREPHOTMEAL}\sqsubseteq\text{PREPAREMEAL}\sqcap\exists\text{REQUIRESUSAGEOFARTIFACT.}$$
$$\big(\text{COOKINGINSTRUMENT}\sqcap(\exists\text{ISIN.}\{\text{APT}\})\big).$$

Thus, we infer which sensor events must necessarily be observed during the execution of an activity. The following axiom states that: "if an event of type *et* is produced by a sensor that detects the usage of an artifact required for executing an activity of class *ac*, then *et* is a *necessary sensor event type* for each activity instance of class *ac*".

$$\text{PRODUCESEVENT}^{-}\circ\text{SENSESUSAGEOF}\circ$$
$$\text{REQUIRESUSAGEOF}^{-}\rightarrow\text{NECESSARYEVENTFOR}.$$

Then, we infer the fillers of property NECESSARYEVENTFOR through ontological reasoning, translate them in MLN_{NC} axioms, and add them to the MLN_{NC} model.

EXAMPLE 6. *Given the introduced axioms, in this case the OWL 2 reasoner infers that {*ET_STOVE*} is a necessary sensor event type for {*AC_PREP_HOT_MEAL*}. Indeed,* ET_STOVE *is produced by usage of* STOVE*, which is the only instance of* COOKINGINSTRUMENT *available in the home.*

RECOGNIZING ACTIVITY INSTANCES

At first, we identify activity instance *candidates* and consider them as part of our MLN_{NC} knowledge base (KB). The KB also includes observed sensor events and computed semantic

Algorithm 1: Statistical analysis of events

Input: Sensor events $\{event(se_0,et_0,t_0),\ldots,event(se_n,et_n,t_n)\}$,
prior probability matrix *PPM*
Output: Candidate activity instances $\{i_0,i_1,\ldots,i_{m-1}\}$

```
instances ← ∅
for each event(se,et,t) ∈ X do
    ac ← activity class with max correlation with et according to PPM
    ai ← activity instance in instances of class ac closest to se
    if ai exists and t is temporally close to ai according to maxGap_ac then
        assign event(se,et,t) to ai
    else
        ai ← a new instance of class ac
        assign event(se,et,t) to ai
        instances ← instances ⋃ {ai}
    end if
end for
return instances
```

correlations. Then, MAP inference enables us to assign each activity instance to its most probable class, and each event to its most probable activity instance. Figure 5 depicts our MLN_{NC} model, where we distinguish between *observed* (star symbol) and *hidden* predicates. Observed predicates represent knowledge facts, where the instances of hidden predicates are computed by MAP inference. In the following, we explain the different components of our framework in detail.

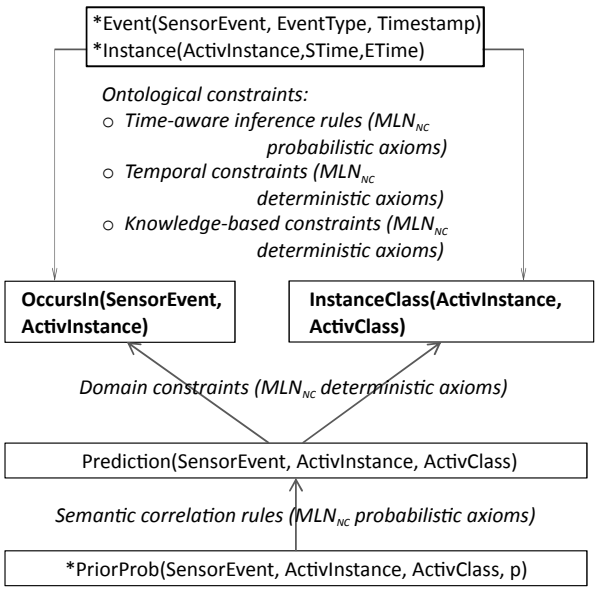

Figure 5. Probabilistic activity recognition framework. The arrows indicate the relations and dependencies between the depicted observed and hidden predicates.

Statistical analysis of events

Candidate activity instances are computed by a heuristic algorithm, shown in Algorithm 1, which implements the STATISTICAL ANALYSIS OF EVENTS module. The algorithm iterates over all temporally ordered events provided by the SEMANTIC INTEGRATION layer. It considers the *PPM* matrix of semantic correlations to infer, for each sensor event *se*, the most probable activity class *ac* generating it. The corresponding timestamp of the event and the resulting activity class enables us to formulate initial hypotheses about the occurred activity instances. If an activity instance *ai* of class *ac* exists, whose boundaries (start and end time) are temporally close to *se* according to an activity-dependent threshold $maxGap_{ac}$, then *se* is assigned to *ai*. Otherwise, a new instance of class *ac* is created, and *se* is assigned to it. The boundaries of each instance are respectively represented by the first and the last event of the instance.

MLN modeling

Semantic correlations are modeled through predicates *PriorProb*, *Event*, and *Instance*. The *PriorProb* predicate represents correlations among sensor events and activities:

$$*PriorProb(SensorEvent, ActivInstance, ActivClass, p)$$

Hence, it describes the probability p that a given sensor event *se* corresponds to a given activity instance *ai* of an activity class *ac*. The probability relies on the semantic correlation between the event type *et* and the activity class *ac* (PPM), and also depends on the temporal distance between the sensor event *se* and the boundaries of the activity instance *ai*.

Formally, given an activity instance *ai* of class *ac* with start time t_{st} and end time t_{ed}, and a sensor event *se* of type *et* and timestamp t, the probability p of $*PriorProb(se,ai,ac,p)$ is computed by the following function:

$$p = \begin{cases} PPM(ac,et) & \text{if } t_{ed}\text{-}MaxGap_{ac} \leq t \leq t_{st}\text{+}MaxGap_{ac} \\ 0 & \text{otherwise} \end{cases}$$

Each sensor event is represented by an instance of the predicate *Event*, which represents the sensor event, its type, and its timestamp:

$$*Event(SensorEvent, EventType, Timestamp)$$

Candidate activity instances computed by Algorithm 1 are represented by the predicate *Instance* which models the relation between the activity instance, its start time, and end time:

$$*Instance(ActivInstance, STime, ETime)$$

The instantiated predicates, derived from the activity instances and the recorded sensor events, are added as facts to our MLN_{NC} knowledge base.

Hidden predicates and domain constraints

Beside the observed predicates, the model also comprises a set of hidden predicates, which can be considered our target classes: *Prediction*, *OccursIn*, and *InstanceClass*. The predicate *Prediction* represents the predicted assignment of a sensor event to an activity instance of a given class:

$$Prediction(SensorEvent, ActivInstance, ActivClass)$$

In addition, the other two predicates are used to express domain constraints about the consistency of inferred activity instances:

$$OccursIn(SensorEvent, ActivInstance)$$
$$InstanceClass(ActivInstance, ActivClass)$$

In particular, the following domain constraint states that each sensor event occurs in exactly one activity instance:

$$|ai|OccursIn(se,ai) = 1,$$

while the following one states that each activity instance belongs to exactly one activity type:

$$|ac|InstanceClass(ai,ac) = 1.$$

Semantic correlation rules

The relations between the observed and hidden predicates are modeled by probabilistic axioms. As illustrated in Figure 5, the hidden predicate *Prediction* is derived from *PriorProb*:

$$conf : *PriorProb(se,ai,ac,conf) \Rightarrow Prediction(se,ai,ac).$$

Thus, the confidence value describes the probability that a sensor event is assigned to an activity instance of a given class. In turn, the remaining hidden predicates are derived from the hidden *Prediction* predicate. The corresponding probabilistic axioms are the following:

$$Prediction(se,ai,ac) \Rightarrow OccursIn(se,ai),$$
$$Prediction(se,ai,ac) \Rightarrow InstanceClass(ai,ac).$$

Note that the above rules are subject to the domain constraints introduced before.

Knowledge-based constraints

Knowledge-based constraints enable us to express conditions about the occurrence (or non-occurrence) of sensor events of a given type during the occurrence of an activity instance.

EXAMPLE 7. *The constraint "each activity instance of type 'preparing hot meal' must be associated to an event of type 'UseStove' " is logically expressed by the rule:*

$$InstanceClass(ai, \text{“}PreparingHotMeal\text{”}) \Rightarrow \exists\, se,t :$$
$$OccursIn(se,ai) \wedge *Event(se, \text{”}UseStove\text{”}, t).$$

Knowledge-based constraints are automatically derived from the fillers of the NECESSARYEVENTFOR OWL 2 property obtained from ontological reasoning as already mentioned.

Temporal constraints

We model MLN_{NC} temporal constraints regarding the duration and the distance of events or activities. We consider two kinds of temporal constraints:

1) *Temporally close events (e.g., whose temporal distance is below Δ seconds) likely belong to the same activity instance.* We express this soft constraint through these axioms:

$$\forall t_1,t_2 : (|t_1 - t_2| < \Delta) \Rightarrow tClose(t_1,t_2)$$

$$w\ Event(se_1,et_1,t_1) \wedge Event(se_2,et_2,t_2) \wedge$$
$$tClose(t_1,t_2) \wedge OccursIn(se_1,ai) \Rightarrow OccursIn(se_2,ai)$$

The latter is a probabilistic axiom whose weight w is chosen experimentally.

2) *Constraints on typical duration of each activity (e.g., "showering cannot last more than Δ' minutes").* We express these constraints either through probabilistic or deterministic axioms, according to the characteristics of the considered activity. Indeed, the variance of the duration of certain activities (e.g., showering) is relatively small, while it is larger for other activities (e.g., preparing dinner). The duration of the former is modeled with deterministic axioms where probabilistic ones are used for the latter. The axioms below state that an instance of "showering" cannot last more than Δ' minutes:

$$\forall t_1,t_2 : (|t_1 - t_2| < \Delta') \Rightarrow tclose\_showering(t_1,t_2)$$

$$InstanceClass(ai, \text{“}Showering\text{”}) \wedge OccursIn(se_1,ai) \wedge$$
$$OccursIn(se_2,ai) \wedge Event(se_1,et_1,t_1) \wedge$$
$$Event(se_2,et_2,t_2) \Rightarrow tclose\_showering(t_1,t_2)$$

Time-aware inference rules

Finally, as explained before, the semantics of some simple activities is naturally expressed in our ontology based on the typical actions composing them. Hence, we apply rules that express the relation of specific operations derived from sensor events in context of time. Consider the following example:

EXAMPLE 8. *A typical pattern of operations for watering plants consists in (1) "getting water" and (2) "moving to the plants" shortly after. We express this activity inference pattern through the MLN_{NC} axioms below:*

$$Event(se_1, \text{“}water\_sensor\text{”}, t_1)$$
$$\wedge Event(se_2, \text{“}plant\_presence\_sensor\text{”}, t_2) \wedge t_1 < t_2$$
$$\wedge tclose\_waterplants(t_1,t_2) \Rightarrow \exists ai :$$
$$InstanceClass(ai, \text{“}WaterPlants\text{”})$$
$$\wedge occursIn(se_1,ai) \wedge occursIn(se_2,ai).$$

Inference of activity instances and temporal boundaries

In order to infer activity instances, their class, and corresponding sensor events, we execute MAP inference on the presented MLN_{NC} model. The output of MAP inference is the most probable assignment of (i) sensor events to activity instances (i.e., fillers of the OccursIn predicate), and (ii) activity classes to activity instances (i.e., fillers of the InstanceClass predicate). Since computing the start and end time of activity instances within MLN_{NC} reasoning would be unnecessarily complicated, we post-process the result of MAP inference to detect the temporal boundaries of each activity instance *ai*:

$$STime(ai) = \min\{t : \exists Event(se,et,t) \wedge OccursIn(se,ai)\},$$
$$ETime(ai) = \max\{t : \exists Event(se,et,t) \wedge OccursIn(se,ai)\}.$$

EXPERIMENTAL EVALUATION

In the following, we present our experimental setup and results. Due to lack of space, we only present aggregated results of all subjects. However, the individual results for each subject, as well as the MLN_{NC} model and the ontology, are available online[2]. Unless otherwise specified, the presented results rely on the introduced unsupervised approach, where the semantic correlations (PPM matrix) were derived from ontological reasoning. To evaluate the effectiveness of semantic correlations

[2] http://sensor.informatik.uni-mannheim.de/#results2016unsupervised

extracted with our method, we also performed experiments computing the *PPM* from the dataset; more precisely, based on the frequency of the sensors types produced by the different activities. We denote by MLN_{NC} (Ontology) the former method, and by MLN_{NC} (Dataset) the latter. We use the well-known dataset of Cook et al. [31, 7], named *CASAS*, and the dataset presented in [28], called *SmartFaber*. Both datasets include interleaved activities in a smart-home environment. To provide the possibility to reconstruct our approaches and experiments, we provide a REST API and web interface which is publicly available[3] and supports the MLN_{NC} solver.

CASAS Dataset

The CASAS dataset covers interleaved ADLs of twenty-one subjects acquired in a smart home laboratory. Sensors collected data about movement, temperature, use of water, interaction with objects, doors, phone; 70 sensors were used in total. Eight activities were considered: *fill medication dispenser* (ac_1), *watch DVD* (ac_2), *water plants* (ac_3), *answer the phone* (ac_4), *prepare birthday card* (ac_5), *prepare soup* (ac_6), *clean* (ac_7), and *choose outfit* (ac_8). The order and expenditure of time were up to the subject and it was allowed to perform the activities in parallel. During the data collection only one single person was present in the smart home. With our MLN_{NC} (Ontology) method, only 25 out of 70 sensors were used. Indeed, the semantic correlation reasoner excluded the remaining 45 (mostly movement sensors), since they had no significant correlation with the considered activities.

During this experiment, we evaluated how well the considered sensor events could be assigned to the corresponding activity instance, but also the quality of detected activity boundaries.

Table 2 shows that our method outperforms the HMM approach used in [31] in assigning each sensor event to the activity instance that generated it. We observe that we recognize each activity at least equal or better than HMM, except *Clean*. The poor performance in recognizing *Clean* is due to the fact that, in the CASAS dataset, it is characterized by different movement patterns that are only partially captured by our method, especially when semantic correlations are extracted from the ontology. Considering the other activities, the *PPM* generated by ontological reasoning obtains essentially the same performance of the one extracted from the dataset, confirming the effectiveness of our semantic correlation reasoner.

Focusing on the other activities, the experiments show that the interactions with objects are strong indicators of the performed activities. However, inspecting the recognition result in detail, we noticed a few cases in which subjects exhibited strange behaviors; e.g., prepared soup without water or took the phone but did not place a phone call. Especially the latter case is hard to recognize without further information. The former case is probably related to sensor errors.

Figure 6 illustrates the individual results in more detail. It highlights that there are cases where we could not recognize the activities *Answer the phone* and *Clean* at all, but in general the distribution is very similar and narrowed.

[3] http://executor.informatik.uni-mannheim.de

Table 2. CASAS dataset: Results (F_1 measure) of the proposed activity recognition method compared to related work for interleaved activities. *Dataset* (supervised) and *Ontology* (unsupervised) describe the source of semantic correlations (PPM matrix).

Class	HMM [31] (time-shifted)	MLN_{NC} (Dataset)	MLN_{NC} (Ontology)
ac_1	0.656	0.803	**0.848**
ac_2	0.862	**0.882**	0.811
ac_3	0.285	**0.740**	0.720
ac_4	0.589	0.688	**0.723**
ac_5	**0.828**	0.807	0.808
ac_6	0.826	0.873	**0.882**
ac_7	**0.881**	0.781	0.574
ac_8	0.673	**0.904**	0.882
avg.	0.700	**0.810**	0.781

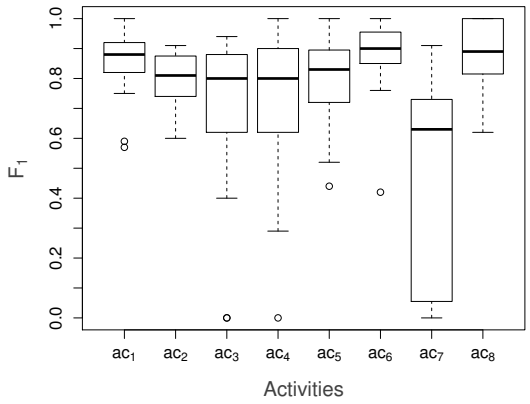

Figure 6. CASAS dataset: Detailed recognition results for each activity, aggregated over all subjects and represented by a box plot.

Considering the boundary detection method, the experiments show that preceding results and the quality of the detected boundaries for the individual activities are weakly related. Table 3 describes the deviation from the actual boundaries in detail. ΔStart is the average difference between the actual and predicted start of an activity instance in minutes. ΔDur is the average difference of actual and predicted duration. In context of the typical duration of each activity, the boundaries are well detected. Hence, the highest deviations are associated with the longest activities, and the overall results are acceptable for most applications.

Table 3. CASAS dataset: Results of boundary detection with MLN_{NC} (Ontology). It shows the average deviation [min] of the candidate compared to the refined instances.

Class	ΔStart (Candidate)	ΔStart (Refined)	ΔDur (Candidate)	ΔDur (Refined)
ac_1	**0.670**	0.765	1.436	**0.890**
ac_2	**0.592**	**0.592**	**2.974**	3.140
ac_3	**0.075**	0.081	0.930	**0.829**
ac_4	**0.079**	**0.079**	**0.341**	0.422
ac_5	1.300	**1.079**	5.810	**4.642**
ac_6	1.617	**0.109**	4.077	**0.803**
ac_7	1.311	**0.692**	2.390	**2.249**
ac_8	**0.079**	0.097	1.300	**0.521**
avg.	0.727	**0.456**	2.424	**1.701**

When we compare the candidate instances and the refined (final) results obtained through MLN_{NC} reasoning, it strikes that our method refines the candidates reliably. Regarding *watch DVD* (ac_2) and *answer the phone* (ac_4), the refined

duration increased slightly, because in some cases subjects took the phone well before using it, or turned on the DVD player well before watching a DVD. Besides, the low numbers clearly show that the duration of the different activities was in general short.

SmartFaber Dataset

The dataset was acquired from an elderly woman diagnosed with Mild Cognitive Impairment, living alone in her apartment. Different environmental sensors (magnetic sensors, motion sensors, temperature sensors) have been used to monitor three ADLs for 55 days: *Taking medicines* (ac_9), *Cooking* (ac_{10}), *Eating*. Moreover, activity *Others* (ac_{11}) was also labeled. Totally, 11 sensors were deployed. Our semantic correlation reasoner discarded 2 sensors among them, because they had no significant correlation with the considered activities. Compared to CASAS, this dataset was acquired in a fully naturalistic environment. Due to the cognitive decline of the subject, activities have been performed in many different and sometimes unexpected ways. Besides, the acquired data is also affected by noise due to various technical issues encountered during data acquisition [28]. Hence, the recognition of ADLs in this scenario is challenging, even if the number of considered activities is limited.

In order to be comparable with the results of previous works on the same dataset, we focused on activity instance classification. Table 4 shows the corresponding results and indicates that the accuracy achieved by our unsupervised method is comparable to the one achieved by the supervised method used in [29]. That method relied on temporally-based feature extraction and on a random forest classifier. However, we were unable to recognize *Eating*, because in the dataset it was only characterized by a single presence sensor close to the table, that was also used in context of the other activities. Hence, our semantic correlation reasoner did not find any sensor significantly correlated to *Eating*. Therefore, we decided to exclude that activity from the evaluation. On the other side, we were able to recognize *Others*, which was not considered in [29].

Inspecting the results, we notice that, with *Cooking*, our unsupervised method achieves essentially the same recognition rate of the supervised technique. With *Taking medicines*, the accuracy of our method is lower, mainly due to the absence of sensors strongly correlated to that activity. The accuracy of recognizing *Others* is in line with the one of the other activities. Considering the corresponding instance boundary results, Table 5 shows that, also with this dataset, MLN_{NC} refinement significantly improves the accuracy of predicted activity instances.

Table 4. SmartFaber dataset: Results (F_1 measure) of the proposed activity recognition method compared to related work. *Dataset* (supervised) and *Ontology* (unsupervised) describe the source of semantic correlations (PPM matrix).

Class	SmartFABER [29] (supervised)	MLN_{NC} (Dataset)	MLN_{NC} (Ontology)
ac_9	**0.946**	0.837	0.831
ac_{10}	**0.757**	0.669	0.752
ac_{11}	-	0.665	**0.702**

Table 5. SmartFaber dataset: Results of the boundary detection method. It shows the average deviation [min] of the candidate compared to the refined instances.

Class	ΔStart (Candidate)	ΔStart (Refined)	ΔDur (Candidate)	ΔDur (Refined)
ac_9	**2.199**	2.533	**1.084**	**1.084**
ac_{10}	14.437	**8.954**	25.833	**21.133**
ac_{11}	7.559	**3.255**	34.170	**16.590**
avg.	8.065	**4.914**	20.362	**12.936**

DISCUSSION AND FUTURE WORK

In this paper, we proposed an unsupervised method to recognize complex ADLs through ontological and probabilistic reasoning. Extensive experiments with real-world datasets showed that the accuracy of our unsupervised method is comparable to the one of supervised approaches, even using a smaller number of sensors.

On the negative side, our technique requires a relevant knowledge engineering effort to define a comprehensive ontology of activities, home environment, and sensor events. For instance, our ontology includes 235 classes and 59 properties. One could argue that the advantage of ontological reasoning does not worth the effort, since it would be easy to manually estimate correlations among activities and sensor events based on common sense. However, consider the CASAS setup used in our experiments: it involves 70 sensors and 8 activities, resulting in 560 combinations of activities and sensor events. Other real-world deployments are much more complex. Hence, manual modeling would be unfeasible in a realistic scenario.

We point out that the knowledge engineering effort can be reduced by reusing existing ontologies. In particular, the ontology used in this work is an extension of the COSAR ontology [26], which was originally intended to model context data and human activities. The extension mainly regarded the definition of a few classes for activities and artifacts that were not considered before, and a few additional properties used by our reasoning method. Developing the extension required one day of work by a researcher with good skills in OWL 2 modeling. Moreover, we were able to use the same ontology for both apartments involved in our experimentation, which had very different characteristics. However, it is questionable whether in larger scale implementations the same ontology can be adequate to cover every possible home environment and individuals' mode of activity execution. We intend to perform extensive experiments in real-world environments to answer this question. Moreover, as a future research direction, we want to exploit active learning to fine-tune the probabilistic model according to the user's environment and personal habits, and to automatically evolve the ontology according to the current context.

Acknowledgement

The authors would like to thank Melisachew Chekol and Jakob Huber for their work on extending Markov Logic with numerical constraints that has proven to be very useful for the work reported in this paper, and Christian Meilicke and Claudio Bettini for fruitful discussions on the problem formulation.

REFERENCES

1. Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. 2010. *The Description Logic Handbook: Theory, Implementation and Applications* (2nd ed.). Cambridge University Press, New York, NY, USA.
2. Ling Bao and Stephen S. Intille. 2004. Activity Recognition from User-Annotated Acceleration Data. In *Pervasive Computing: Second International Conference, PERVASIVE 2004, Linz/Vienna, Austria, April 21-23, 2004. Proceedings*. Springer, Berlin, Heidelberg, 1–17.
3. Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A Tutorial on Human Activity Recognition Using Body-worn Inertial Sensors. *ACM Computing Surveys (CSUR)* 46, 3 (2014), 33:1–33:33.
4. Alberto Calatroni, Daniel Roggen, and Gerhard Tröster. 2011. Collection and curation of a large reference dataset for activity recognition. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*. IEEE, Anchorage, Alaska, USA, 30–35.
5. Melisachew Chekol, Jakob Huber, Christian Meilicke, and Heiner Stuckenschmidt. 2016. Markov Logic Networks with Numerical Constraints. In *22st European Conference on Artificial Intelligence (ECAI2016)*. IOS Press, Amsterdam, The Netherlands, 1–9.
6. Liming Chen and Chris Nugent. 2009. Ontology-based activity recognition in intelligent pervasive environments. *International Journal of Web Information Systems* 5, 4 (2009), 410–430.
7. Diane J. Cook, Aaron S. Crandall, Brian L. Thomas, and Narayanan C. Krishnan. 2013a. CASAS: A Smart Home in a Box. *Computer* 46, 7 (2013), 62–69.
8. Diane J. Cook, Kyle D. Feuz, and Narayanan Chatapuram Krishnan. 2013b. Transfer learning for activity recognition: A survey. *Knowledge and Information Systems* 36, 3 (2013), 537–556.
9. Nigel Davies, Daniel P. Siewiorek, and Rahul Sukthankar. 2008. Activity-Based Computing. *IEEE Pervasive Computing* 7, 2 (2008), 20–21. DOI: http://dx.doi.org/10.1109/MPRV.2008.26
10. Prafulla Dawadi, Diane J. Cook, and Maureen Schmitter-Edgecombe. 2013. Automated Cognitive Health Assessment Using Smart Home Monitoring of Complex Tasks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 43, 6 (2013), 1302–1313.
11. Christopher W. Geib and Robert P. Goldman. 2009. A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence* 173, 11 (2009), 1101–1132.
12. Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter F. Patel-Schneider, and Ulrike Sattler. 2008. OWL 2: The next step for OWL. *Journal of Web Semantics* 6, 4 (2008), 309–322.
13. Tao Gu, Zhanqing Wu, XianPing Tao, Hung Keng Pung, and Jian Lu. 2009. epSICAR: An Emerging Patterns based Approach to Sequential, Interleaved and Concurrent Activity Recognition. In *Proceedings of the Seventh Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE Computer Society, Washington, D.C., 1–9.
14. Rim Helaoui, Daniele Riboni, and Heiner Stuckenschmidt. 2013. A Probabilistic Ontological Framework for the Recognition of Multilevel Human Activities. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, New York, NY, USA, 345–354.
15. Xin Hong, Chris D. Nugent, Maurice D. Mulvenna, Suzanne Martin, Steven Devlin, and Jonathan G. Wallace. 2012. Dynamic similarity-based activity detection and recognition within smart homes. *International Journal of Pervasive Computing and Communications* 8, 3 (2012), 264–278.
16. Jakob Huber, Christian Meilicke, and Heiner Stuckenschmidt. 2014. Applying markov logic for debugging probabilistic temporal knowledge bases. In *AKBC 2014: 4th Workshop on Automated Knowledge Base Construction AKBC 2014 at NIPS 2014 in Montreal, Canada, December 13, 2014*. ACM, New York, NY, USA, 1–6.
17. Jonathan Lester, Tanzeem Choudhury, Nicky Kern, Gaetano Borriello, and Blake Hannaford. 2005. A Hybrid Discriminative/Generative Approach for Modeling Human Activities. In *Proceedings of the 19th international joint conference on Artificial intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 766–772.
18. Seng W. Loke. 2004. Representing and reasoning with situations for context-aware pervasive computing: A logic programming perspective. *The Knowledge Engineering Review* 19, 3 (2004), 213–233.
19. Paul Lukowicz, Jamie A. Ward, Holger Junker, Mathias Stäger, Gerhard Tröster, Amin Atrash, and Thad Starner. 2004. Recognizing Workshop Activity Using Body Worn Microphones and Accelerometers. In *Pervasive Computing: Second International Conference, PERVASIVE 2004, Linz/Vienna, Austria, April 21-23, 2004. Proceedings*. Springer, Berlin, Heidelberg, 18–32.
20. Georgios Meditskos, Efstratios Kontopoulos, and Ioannis Kompatsiaris. 2014. Knowledge-Driven Activity Recognition and Segmentation Using Context Connections. In *The Semantic Web – ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part II*. Springer, Cham, 260–275.
21. Georgios Meditskos, Efstratios Kontopoulos, and Ioannis Kompatsiaris. 2015. ReDef: Context-aware Recognition of Interleaved Activities using OWL 2 and Defeasible Reasoning. In *Joint Proceedings of SSN-TC and OrdRing*

2015 (CEUR Workshop Proceedings), Vol. 1488. CEUR-WS.org, Bethlehem, Pennsylvania, United States, 31–42.

22. George Okeyo, Liming Chen, Hui Wang, and Roy Sterritt. 2014. Dynamic sensor data segmentation for real-time knowledge-driven activity recognition. *Pervasive and Mobile Computing* 10, Part B (2014), 155–172.
23. Paulito Palmes, Hung Keng Pung, Tao Gu, Wenwei Xue, and Shaxun Chen. 2010. Object relevance weight pattern mining for activity recognition and segmentation. *Pervasive and Mobile Computing* 6, 1 (2010), 43 – 57.
24. Carolyn Parsey and Maureen Schmitter-Edgecombe. 2013. Applications of technology in neuropsychological assessment. *The Clinical Neuropsychologist* 27, 8 (2013), 1328–1361.
25. Attila Reiss and Didier Stricker. 2012. Creating and benchmarking a new dataset for physical activity monitoring. In *Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments*. ACM, New York, NY, USA, 40:1–40:8.
26. Daniele Riboni and Claudio Bettini. 2011a. COSAR: Hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing* 15, 3 (2011), 271–289.
27. Daniele Riboni and Claudio Bettini. 2011b. OWL 2 modeling and reasoning with complex human activities. *Pervasive and Mobile Computing* 7, 3 (2011), 379–395.
28. Daniele Riboni, Claudio Bettini, Gabriele Civitarese, Zaffar Haider Janjua, and Viola Bulgari. 2015. From Lab to Life: Fine-grained Behavior Monitoring in the Elderly's Home. In *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on*. IEEE Computer Society, Washington, D.C., 342–347.
29. Daniele Riboni, Claudio Bettini, Gabriele Civitarese, Zaffar Haider Janjua, and Rim Helaoui. 2016. SmartFABER: Recognizing Fine-grained Abnormal Behaviors for Early Detection of Mild Cognitive Impairment. *Artificial Intelligence in Medicine* 67 (2016), 57–74.
30. Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine learning* 62, 1 (2006), 107–136.
31. Geetika Singla, Diane J Cook, and Maureen Schmitter-Edgecombe. 2009. Tracking activities in complex settings using smart environment technologies. *International journal of biosciences, psychiatry, and technology (IJBSPT)* 1, 1 (2009), 25–35.
32. Timo Sztyler and Heiner Stuckenschmidt. 2016. On-body Localization of Wearable Devices: An Investigation of Position-Aware Activity Recognition. In *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE Computer Society, Washington, D.C., 1–9.
33. X. H. Wang, T. Gu, D. Q. Zhang, and H. K. Pung. 2004. Ontology Based Context Modeling and Reasoning using OWL. In *Proceedings of Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*. IEEE Computer Society, Washington, D.C., 18–22.
34. Danny Wyatt, Matthai Philipose, and Tanzeem Choudhury. 2005. Unsupervised Activity Recognition Using Automatically Mined Common Sense. In *Proceedings of the 20th National Conference on Artificial Intelligence*, Vol. 1. AAAI Press, California, USA, 21–27.
35. Juan Ye, Simon Dobson, and Susan McKeever. 2012. Situation identification techniques in pervasive computing: A review. *Pervasive and Mobile Computing* 8, 1 (2012), 36–66.
36. Juan Ye and Graeme Stevenson. 2013. *Semantics-Driven Multi-user Concurrent Activity Recognition*. Springer International Publishing, Cham, 204–219.
37. Juan Ye, Graeme Stevenson, and Simon Dobson. 2014. USMART: An Unsupervised Semantic Mining Activity Recognition Technique. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 4, 4 (2014), 16:1–16:27.