Personal and Ubiquitous Computing manuscript No.
(will be inserted by the editor)

COSAR: Hybrid Reasoning for Context-aware Activity Recognition

Daniele Riboni · Claudio Bettini

the date of receipt and acceptance should be inserted later

Abstract Human activity recognition is a challenging problem for context-aware systems and applications. Research in this field has mainly adopted techniques based on supervised learning algorithms, but these systems suffer from scalability issues with respect to the number of considered activities and contextual data. In this paper, we propose a solution based on the use of ontologies and ontological reasoning combined with statistical inferencing. Structured symbolic knowledge about the environment surrounding the user allows the recognition system to infer which activities among the candidates identified by statistical methods are more likely to be the actual activity that the user is performing. Ontological reasoning is also integrated with statistical methods to recognize complex activities that cannot be derived by statistical methods alone. The effectiveness of the proposed technique is supported by experiments with a complete implementation of the system using commercially available sensors and an Android-based handheld device as the host for the main activity recognition module.

1 Introduction

There is a general consensus on the need for effective automatic recognition of user activities to enhance the ability of a pervasive system to properly react and adapt to the circumstances. Among many applications of activity recognition, a special interest is in the pervasive e-Health domain where automatic activity recognition is used in rehabilitation systems, chronic disease management, monitoring of the elderly, as well as in personal well being applications (see, e.g., [6, 26, 2]).

Example 1 Consider the case of Alice, an elderly person undergoing rehabilitation after having been hospitalized for a minor heart attack. In order to help Alice in correctly following the practitioners' prescriptions about the physical activities to perform during rehabilitation, the hospital center provides her with a monitoring system that continuously keeps track of her physiological data as well as of her activities. In particular, physiological data (e.g., heart rate and blood pressure) are acquired by wearable sensors that transmit them through a wireless link to the monitoring application hosted on her mobile phone. Similarly, accelerometer data provided by a fitness watch are transmitted to the monitoring application and merged with those provided by the accelerometer integrated in her mobile phone to automatically infer her current physical activity. On the basis of physiological data and performed activities, the monitoring application provides Alice with alerts and suggestions to better follow her rehabilitation plan (e.g., "please consider to take a walk this morning", or "take some rest now"). Moreover, those data are reported to the medical center on a daily basis for further processing.

Of course, for such a system to be effective, the activity recognition module must provide very accurate results. In fact, if activities are wrongly recognized, the monitoring system may draw erroneous conclusions about the actual adherence of the patient to the practitioners' prescriptions, as well as provide error-prone statistics about the health status of the patient.

D. Riboni
Università degli Studi di Milano, D.I.Co.
E-mail: riboni@dico.unimi.it

C. Bettini
Università degli Studi di Milano, D.I.Co.
E-mail: bettini@dico.unimi.it

A huge amount of research work has been done on techniques for activity recognition, and the prominent approaches in context-awareness are briefly presented in Section 2. We note that the most effective of these systems currently rely on the application of supervised learning algorithms. In order to provide good performance, these algorithms need to be trained with a sufficiently large amount of labeled data. Indeed, the use of a small set of training data, in presence of a wide set of context data, would be ineffective, if not counterproductive, since the classifier could draw erroneous predictions due to the problem of overfitting. For instance, in [17] some available context data are discarded in order to avoid this problem, that turns out to be one of the main reasons why activity recognition systems do not perform well out of the laboratory. Since training data are very hard to acquire, systems relying on supervised learning are prone to serious scalability issues the more activities and the more context data are considered. For example, suppose to consider as the only context data the user's current symbolic location (e.g., *kitchen*, *dining room*, *mall*, *park*, etc). Even in this simple case, in order to gain good recognition results a sufficiently large set of training data should be acquired for each activity in any considered location. Of course, such a large set of training data is very hard to obtain. Moreover, when we consider as context not only location but also environmental conditions and surrounding objects, the task of collecting a sufficient amount of training data is very likely to become unmanageable, since training data should be acquired in any possible contextual condition.

In this paper we investigate the use of ontological reasoning coupled with statistical reasoning in order to address the above-mentioned problem. The intuition behind our solution is the following. Statistical inferencing is performed based on raw data retrieved from body-worn sensors (e.g., accelerometers) to predict the most probable activities. Then, symbolic reasoning is applied to refine the results of statistical inferencing by selecting the set of possible activities performed by a user based on her current context. For a large class of simple activities, the influence of context on the raw data gathered from sensors is minimal; hence, the same statistical model can be applied. Only when context strongly affects the data values from sensors, it is necessary to acquire training data under different context conditions. Hence, context information is mostly exploited by the ontological reasoner, and only in limited cases by the statistical reasoner. By decoupling the use of context information, statistical inferencing becomes more manageable in terms of necessary training data, while symbolic reasoning can more effectively select candidate activities taking into account context-dependent ontological relationships. In order to perform context-based symbolic reasoning, we have defined an ontology that models activities, artifacts, persons, communication routes, and symbolic locations, and that expresses relations and constraints among these entities. The same ontology is used to describe and recognize complex social activities that would be hardly identifiable by a purely statistical technique. To the best of our knowledge this is the first work (except our preliminary results presented in [23]) that systematically investigates the integration of statistical and ontological reasoning for activity recognition.

In summary, these are the main contributions of this paper:

- We define an architecture for a mobile context-aware activity recognition system (COSAR) supporting hybrid statistical and ontological reasoning; The architecture includes an ontology of human activities.
- We propose a new variant of multiclass logistic regression as the statistical recognition method, and we design an algorithm to integrate this method with ontological reasoning.
- We illustrate a complete implementation of the core modules of the architecture, including the ones for sensors and mobile devices; We show the effectiveness of the COSAR system and its superiority with respect to a purely statistical method by testing the implementation on a (publicly available) set of data collected from real users.

The rest of the paper is organized as follows: Section 2 discusses related work; Section 3 presents the architecture of the COSAR activity recognition system; Section 4 presents the basic techniques used in COSAR for statistical and ontological reasoning, while Section 5 technically describes how they are combined in the overall recognition algorithm; Section 6 explains how the main modules have been implemented and Section 7 presents the experimental results; Section 8 concludes the paper.

2 Related work

Many techniques have been proposed to automatically recognize human activities based on different kinds of data. The main approaches consist in the use of either statistical or symbolic reasoning. However, up to the time of writing, these approaches have mainly been considered separately.

Proposed statistical activity recognition techniques differ on the kind and number of used sensors, considered activities, adopted learning algorithms, and many other parameters. A research direction consists in the use of cameras with the help of sound, image and scene recognition software (see, e.g., [21, 5, 28]). While these techniques can be profitably exploited in particular scenarios (surveillance systems, smart-room and smart-office applications, ...), in general their applicability is limited to confined environments, and they are often subject to serious privacy concerns, clearly perceived by the monitored users.

Alternative activity recognition techniques are based on data acquired from body-worn sensors (e.g., motion tracking and inertial sensors, cardiofrequencimeters, ...) and on the application of statistical learning methods. Early attempts in this sense were mainly based on the use of data acquired from multiple body-worn accelerometers (e.g., [8, 15]). One of the main limitations of these early systems relied on the fact that they did not consider contextual information (such as current location, environmental conditions, surrounding objects) that could be usefully exploited to derive the user's activity. As a consequence, later approaches were aimed at devising activity recognition systems taking into account the user's context. For instance, in [17] a method is proposed to classify physical activities by considering not only data retrieved from a body-worn accelerometer, but also environmental data acquired from several other sensors (sound, humidity, acceleration, orientation, barometric pressure, ...). Spatio-temporal traces are used in [18] to derive high-level activities such as *shopping* or *dining out.* Observations regarding the user's surrounding environment (in particular, objects' use), possibly coupled with body-worn sensor data, are the basis of many other activity recognition systems (e.g., [27, 25, 10, 12]). However, as anticipated in the introduction, these systems suffer from serious scalability issues with respect to the number of considered context data. This challenging problem has been addressed (e.g., in [14]) by means of a combination of supervised and unsupervised learning techniques. We argue that, while similar techniques can be adopted to mitigate the problem, it is unlikely that they can provide a definitive solution.

On the other hand, the recognition of complex activities like social ones (e.g., *work meeting*, *friendly chat*) is particularly challenging, and it is hard to be achieved by the use of solely statistical methods. Indeed, complex activities can be better recognized by considering constraints and relationships among context data that neither can be directly acquired from sensors, nor can be derived through statistical reasoning alone. For instance, one possible condition to recognize the activity *giving a class* is the case in which the actor is a teacher, the actor's current location is a classroom, some students are in the classroom, and the actor is writing on a blackboard. For this reason, some researchers have investigated the use of ontologies to represent complex activities (e.g., the ontologies SOUPA [7], CONON [9], and the one of CARE [1]). In order to avoid the limitations of solely ontological reasoning, such techniques adopt a combination of ontological reasoning and logic programming. However, the recognition of complex activities through ontological reasoning has to start from some basic observations (e.g., the user is in a given building, he is standing, etc...) and the purely ontology-based approaches do not deeply investigate how this data can be acquired and transformed into the ontology representation. Our proposed technique integrates the recognition of these basic observations through statistical analysis of sensor data with ontological reasoning.

3 COSAR system architecture

The proposed activity recognition system is graphically depicted in Figure 1. The lower layer (SENSORS) includes body-worn sensors (providing data such as accelerometer readings and physiological parameters) and sensors spread in the environment.

Data provided by environmental and body-worn sensors are communicated through a wireless connection to the USER MOBILE DEVICE, and merged with sensor data retrieved by the device itself (e.g., data provided by an embedded accelerometer) to build a *feature vector* that will be used to predict the user's activity. The device also continuously keeps track of the current physical location provided by a GPS receiver. When the GPS reading is not available or not sufficiently accurate (e.g., indoor), localization is performed by an external LOCATION SERVER (e.g., a GSM triangulation system provided by the network operator, or an RFID system). The GIS module is in charge of mapping the physical location reading to the most specific symbolic location that correspond to that physical location. This information will be used by the COMBINED ONTOLOGICAL/STATISTICAL ACTIVITY RECOGNITION module (COSAR) to refine the statistical predictions (Section 5.4).

The INFRASTRUCTURE layer includes a PATTERN RECOGNITION module that is in charge of deriving a statistical model of the considered activities (Section 4.1), which is communicated offline to the COSAR module. This layer is also in charge of performing ontological reasoning to calculate the set of activities that can be potentially performed in a given context (Section 4.2). This set is also communicated offline to the COSAR module. Contextual information such as environmental data and basic activities recognized by the COSAR module

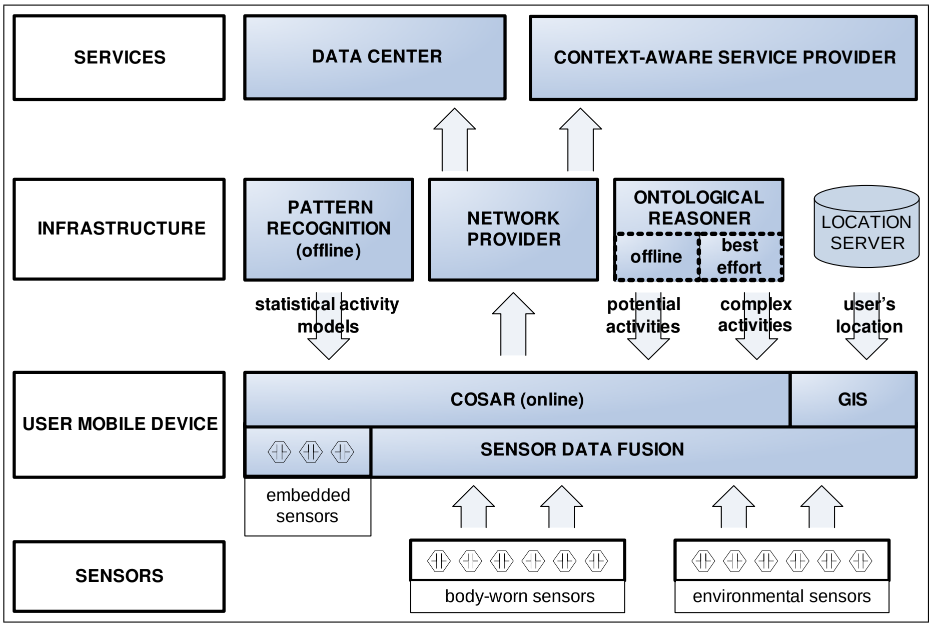


Fig. 1 The *COSAR* system

are communicated on a per-request basis to the ontological reasoner to possibly recognize more complex activities (Section 5.3). In order to cope with the requirements of services needing very fast response times, ontological reasoning to derive complex activities is performed in a *best-effort* fashion; for instance, reasoning is terminated when a predefined time-to-live expires. In addition, the infrastructure layer includes a network provider offering the connectivity necessary to exchange data between modules at different layers, and, in particular, to communicate activity information to remote data centers or context-aware service providers.

With respect to efficiency issues, we point out that the most computationally expensive tasks (i.e., ontological reasoning and pattern recognition to build a statistical model of activities) are executed by servers in the infrastructure domain. Note that privacy issues are of paramount importance in this domain; however, their treatment is outside the scope of this paper. Techniques to integrate privacy preservation in a context-aware middleware can be found in [22] and in [24].

4 Reasoning modules

In this section we present the statistical and symbolical reasoning modules of COSAR. The technique to integrate these modules will be presented in Section 5.

4.1 Statistical classification of activities with a *historical* variant

As illustrated in the introduction, the most common approach to activity recognition is to make use of supervised statistical learning methods. Roughly speaking, these methods rely on a set of preclassified activity instances that are used in a training phase to learn a statistical model of a given set of activities. The obtained model is then used to automatically classify new activity instances. Each activity instance is represented by means of a *feature vector*, in which each feature corresponds to a given measure (typically, a statistics about some measurements retrieved from a set of sensors).

Even if significant exceptions exist (e.g., Hidden Markov Models and Linear Dynamical Systems [4]), it is worth to note that most models adopted by statistical learning algorithms implicitly assume independence between each pair of instances to be classified. As a consequence, the prediction of an instance i_2 does not depend on the prediction of another instance i_1. However, when considering activity instances the above-mentioned assumption does not hold. In fact, persons do not continuously switch among different activities; instead, they tend to perform the same activity for a certain lapse of time before changing activity.

In our technique, we exploit this characteristic to improve the activity recognition rates of statistical learning algorithms by means of a novel *historical* variant. We call *duration* the temporal resolution at which activity instances are considered. The following example explains the rationale of our variant.

Example 2 Let $\overrightarrow{f} = \langle f_1, f_2, f_3, f_4, f_5 \rangle$ be a vector of feature vectors corresponding to five activity instances $\langle i_1, i_2, i_3, i_4, i_5 \rangle$ consecutively performed by a user, each instance having a short duration (e.g., a time window of a few seconds). Suppose $\overrightarrow{f}$ is the input for the baseline (not historical) classifier, and its computed prediction is:

$$\overrightarrow{p} = \langle p_1 = \textit{jogging},\ p_2 = \textit{jogging},\ p_3 = \textit{brushingTeeth},\ p_4 = \textit{jogging},\ p_5 = \textit{jogging} \rangle,$$

meaning that the algorithm predicted that the user was jogging for the first two time windows, then brushing teeth for one, and then jogging again for other two time windows.

From this output it is easy to guess that the prediction for the third activity instance was wrong, and that the correct prediction for that instance was *jogging.*

Errors similar to the one highlighted by Example 2 often occur in real-world situations, because, e.g., it may happen that a person performing a given activity abruptly performs some movement that diverges from the normal activity pattern, thus “confusing” the classifier.

In order to address this problems, our historical variant consists in classifying each activity instance i_j based not only on the prediction p_j of the baseline classifier for i_j, but also on its predictions for k activity instances consecutively performed before i_j (i.e., $i_{j-1}, i_{j-2}, \ldots, i_{j-k}$). Using a metaphor, if we call “votes for i_j” the elements in $\overrightarrow{p_j} = \langle p_j, p_{j-1}, p_{j-2}, \ldots, p_{j-k} \rangle$ (i.e., predictions of the baseline classifier), our variant consists in choosing as the predicted activity for i_j the one having the highest number of votes in $\overrightarrow{p_j}$. In the case in which this activity is not unique, one at random is chosen among the ones having the maximum number of votes.

In the rest of this paper we call *statistical-hist* technique the historical variant of a statistical activity recognition algorithm, and we represent with p'_j the prediction for activity instance i_j computed by the *statistical-hist* technique.

Example 3 Continuing Example 2, suppose to apply the statistical-hist technique to the prediction of activity instance i_3 using $k = 2$. Since the votes for i_3 are $\langle p_1 = \textit{jogging}, p_2 = \textit{jogging}, p_3 = \textit{brushingTeeth} \rangle$, the predicted activity p'_3 is *jogging.*

It is worth to note that, despite the use of a similar *vote* metaphor, our technique is different from *ensembles* (or *classifier committees*). Indeed, while in ensembles the single predictions of multiple algorithms are combined to classify a given instance, in our technique we consider multiple predictions of a single classifier. Our technique is also different from *boosting*, since boosting is applied on the training phase, while our variant is applied only during classification.

Note that the value of the parameter k, as well as the duration of activity instances, must be carefully chosen based on the considered activities. In general, large values of k are well suited to activities that tend to last long, while if the user changes activities more frequently, smaller values of k are more appropriate. For the sake of this paper, we assume the use of a fixed value of k, because we mainly consider physical activities having homogeneous durations. However, our technique can be easily extended by dynamically adapting the value of k to the set of candidate activities. While the variant explained above can apply to a large class of statistical learning techniques, in the experiments described in Section 7 we applied it to a specific technique: multiclass logistic regression.

Finally, we should mention that there are cases in which raw data from sensors are strongly influenced by context. This occurs, for example, when a medical condition of the user influences her movements, and we are using inertial sensors. In these cases, it is necessary to perform statistical inferencing with an appropriate statistical model, which of course requires the acquisition of a dedicated training set. Based on our experiments with accelerometers and with context limited to location, we obtained satisfactory results without the need of considering different statistical models.

4.2 Reasoning with the *ActivO* ontology

Even if ontological reasoning can be applied to any kind of context data, in the rest of this paper we concentrate on location information. Indeed, location is an important case of context information, and the current symbolic location of a user can give useful hints about which activities she can or cannot perform. Moreover, from a practical perspective, localization technologies are more and more integrated in mobile devices and buildings; hence, differently from other context data, location information is available in many situations.

4.2.1 The ActivO ontology

We have defined an OWL-DL [13] ontology (named *ActivO*) for the activity recognition domain. The ontology models a set of activities, as well as context data that can be useful to recognize them. The main classes and properties of the ontology are graphically depicted in Figure 2. The main classes of ActivO are `Activity`,

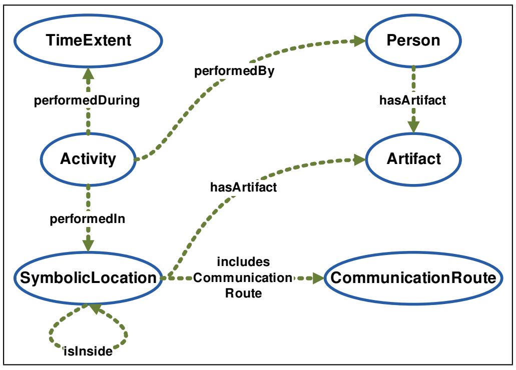


Fig. 2 The core of the *COSAR-ONT* ontology

Class	Descendants
Activity	35
Artifact	43
CommunicationRoute	14
Person	4
SymbolicLocation	30
TimeExtent	11

Table 1 Number of class descendants

SymbolicLocation, CommunicationRoute, Artifact, Person, and TimeExtent. Properties relate instances of those classes. For instance, each activity is isPerformedBy some persons; it is PerformedIn a symbolic location; and it is isPerformedDuring a time extent. The set of activities and context data defined in our ontology is obviously non exhaustive; however, we believe that this ontology can be profitably used to model many pervasive computing scenarios. Moreover, the ontology is easily extensible to address additional application domains.

Figure 3 shows part of the activities modeled by our ontology. Each activity can be either an:

$$\texttt{IndividualActivity} \sqsubseteq \texttt{Activity} \sqcap = 1\, \texttt{hasActor}$$

(individual activities have exactly one actor), or a

$$\texttt{SocialActivity} \sqsubseteq \texttt{Activity} \sqcap \geq 2\, \texttt{hasActor}$$

(social activities have at least 2 actors). Social activities are further classified as Play or Communication. A communication can be either a FaceToFaceMeeting or a CommunicationThroughDevice, and so on.

Individual activities include:

- PersonalActivity, such as BrushingTeeth, Bathing, Eating. These and other activities of daily living are very important in the health care domain, for instance to monitoring the elderly at home;
- Traveling, such as RidingBicycle, MovingByCar, and other activities useful to characterize the situation of people on the move;
- PhysicalActivity (Jogging, Strolling, Hiking, . . .). Monitoring these activities has important applications in the health care and well being domains;
- ProfessionalActivity, having applications in systems for workers support; and
- UsingDevice (UsingPersonalComputer, WritingOnBlackboard, . . .). Recognition of these simple activities can be profitably exploited to recognize more complex activities, as illustrated in Section 5.3.

Since location is very important to recognize activities, particular efforts have been made to model the part of the ontology representing symbolic locations (shown in Figure 4). Each location can be either an IndoorLocation or an OutdoorLocation. Outdoor locations are classified as NonPedestrianOutdoorLocation or PedestrianOutdoorLocation (e.g., Garden, OutdoorSportCenter, UrbanArea, . . .). Several classes have been defined to classify indoor locations. In particular, ActivO models both Buildings (like Hospital, Mall, CampusBuilding), and Rooms (like ConferenceRoom, Office, Laboratory, HospitalRoom, LivingRoom).

Class definitions are characterized according to their relations to other classes of the ontology. For example, locations are characterized according to the kind of Artifact they contain. As it will be shown in Section 4.2.2, this aspect will be exploited to identify potential activities based on the user's current context. For instance, with:

$$\texttt{ConferenceRoom} \sqsubseteq \texttt{Room} \sqcap \exists\, \texttt{hasArtifact.(Projector} \sqcup \texttt{Blackboard} \sqcup \ldots) \quad (1)$$

we state that instances of class ConferenceRoom must contain a device such as a Projector or a Blackboard; while, with:

$$\texttt{HospitalRoom} \sqsubseteq \texttt{Room} \sqcap \exists\, \texttt{isInside.HospitalBuilding} \quad (2)$$

we state that hospital rooms are necessarily located inside a hospital.

4.2.2 Identification of potential activities

Even if in theory the set of possible activities that can be performed in a given symbolic location could be manually specified by a domain expert, this method would be clearly impractical. Indeed, even considering a few tens of activities and symbolic locations, the number of their combinations would quickly render this task unmanageable. Moreover, this task should be repeated each time the characteristics of a symbolic location change (e.g., when an artifact is added to or removed from a room).

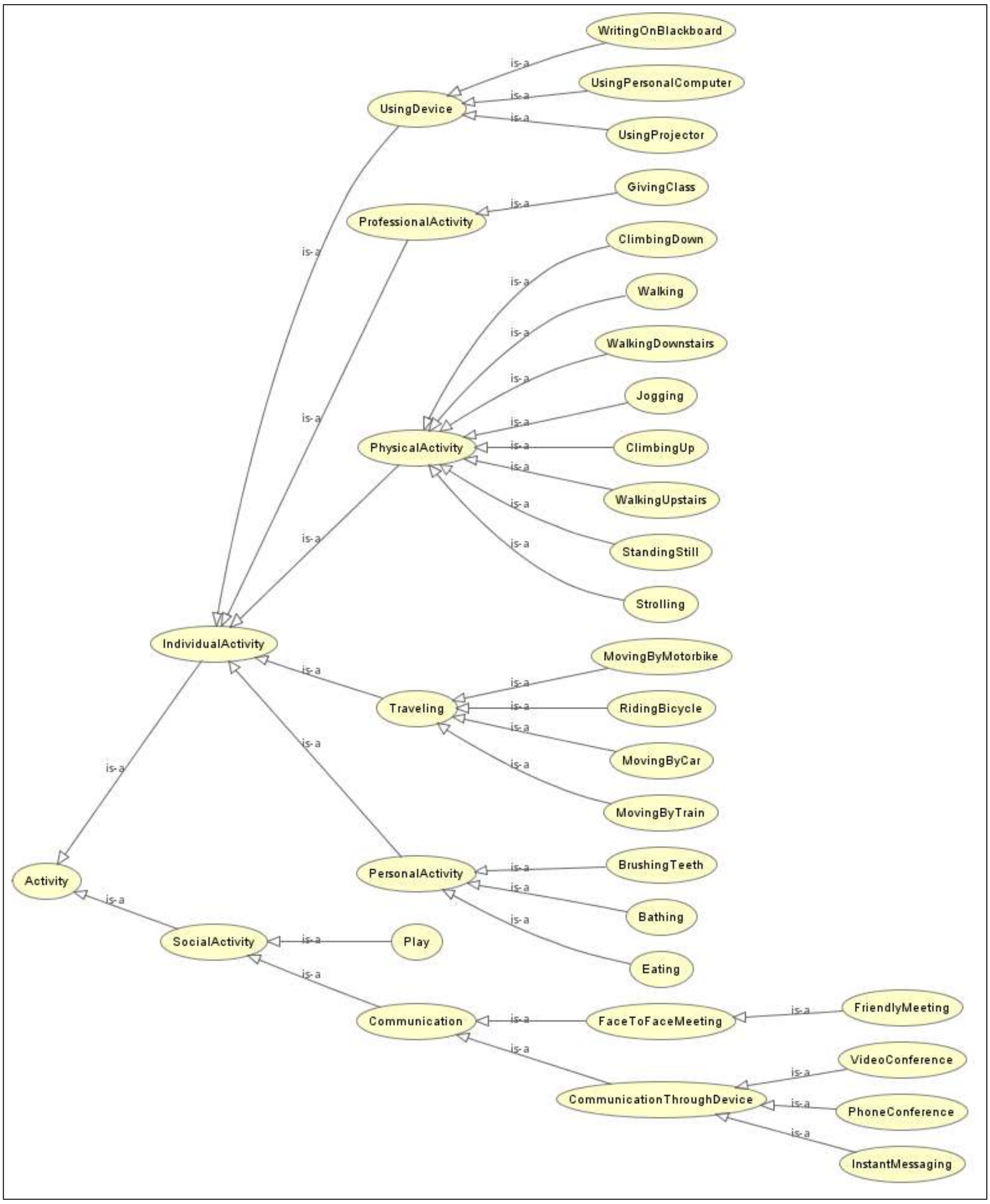


Fig. 3 Part of the ontology of activities

In order to illustrate our technique we introduce the following example.

Example 4 Consider the activity `BrushingTeeth`, and the task of automatically inferring the set of symbolic locations in which such activity can reasonably be performed. One possible definition of the considered activity is the following:

$$\texttt{BrushingTeeth} \sqsubseteq \texttt{PersonalActivity} \sqcap \\ \forall\, \texttt{performedIn}.\,(\,\exists\, \texttt{hasArtifact.Sink}\,) \sqcap \ldots$$

According to the above definition, `BrushingTeeth` is a subclass of `PersonalActivity` that can be performed only in locations that contain a `Sink` (that is defined as a subclass of `WaterFixture`); other restrictions may follow, but they are not considered here for simplicity. Now consider two symbolic locations, namely `RestRoom` and `LivingRoom`, defined as follows:

$$\texttt{RestRoom} \sqsubseteq \texttt{Room} \sqcap \\ \exists\, \texttt{hasArtifact.Sink} \sqcap \ldots$$

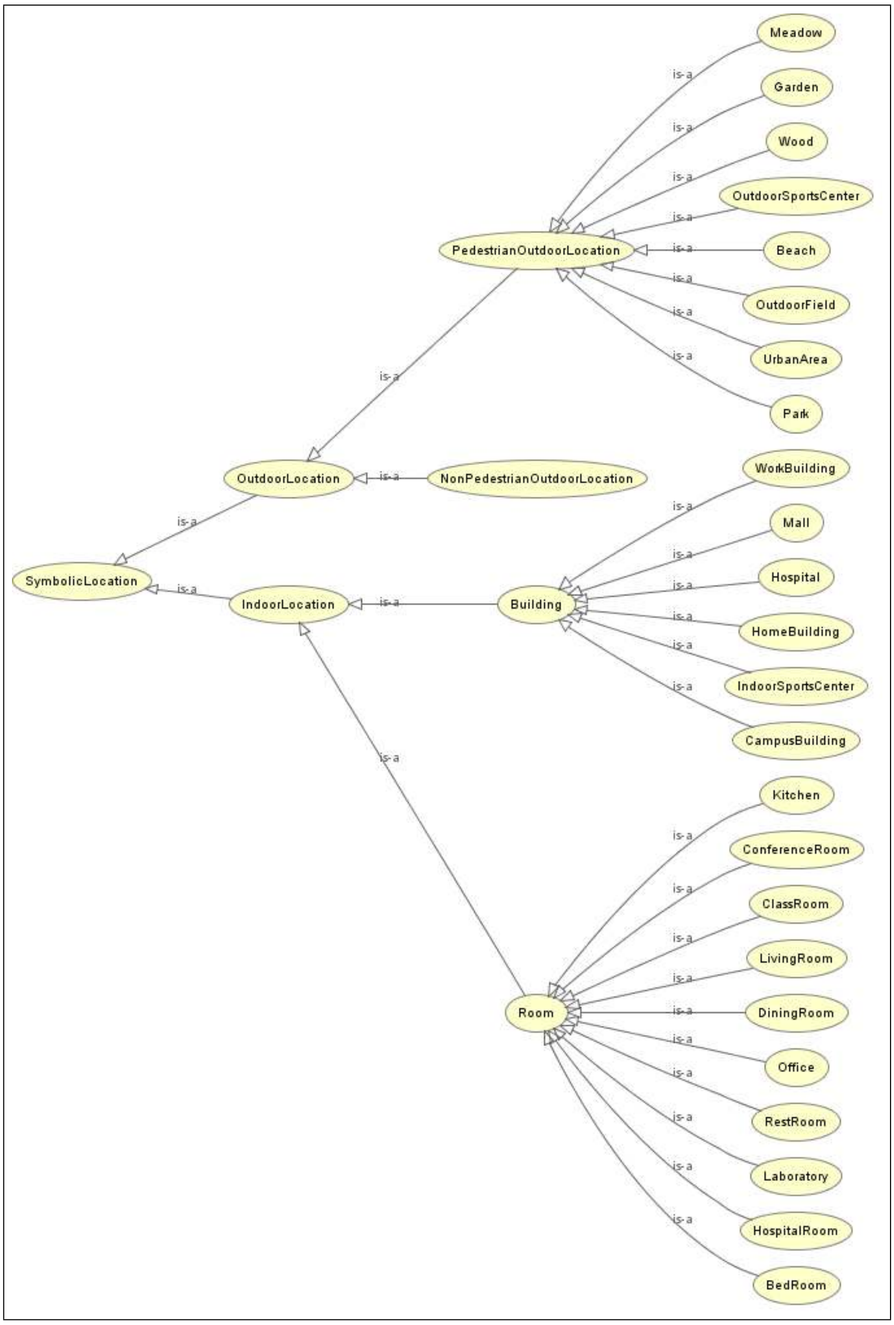


Fig. 4 Part of the ontology of symbolic locations

$$\texttt{LivingRoom} \sqsubseteq \texttt{Room} \sqcap \neg\exists\, \texttt{hasArtifact.WaterFixture} \sqcap \ldots$$

According to the above definitions, `RestRoom` is a `Room` that contains a sink, while `LivingRoom` is a `Room` that does not contain any `WaterFixture` (once again, other details about the definition of these classes are omitted)[1]. Given those ontological definitions it is possible to automatically derive through ontological reasoning the set of symbolic locations in which the activity `BrushingTeeth` can be performed. To this aim, the following assertions are stated and added to the asser-

[1] Note that, due to the open-world assumption of description logic systems [3] and, consequently, of OWL-DL, it is necessary to explicitly state those artifacts that are not present in a given location. This is simplified by considering in the definition of symbolic locations only artifacts that characterize the activities to be discriminated and using the artifact ontology to exclude whole classes of artifacts, as done in the `LivingRoom` example with `WaterFixture`.

tional part of the ontology (called *ABox*):

`BrushingTeeth(CURR_ACT);`

`RestRoom(CURR_LOC_1);`

`LivingRoom(CURR_LOC_2).`

The above assertions create an instance of activity `BrushingTeeth` identified as `CURR_ACT`, an instance of location `RestRoom` identified as `CURR_LOC_1`, and an instance of location `LivingRoom` identified as `CURR_LOC_2`. Then, in order to understand if a given activity instance a can be performed in a given location l it is sufficient to add an assertion to the ABox stating that activity a is performed in location l, and then to check if the ABox is consistent with respect to the terminological part of the ontology by performing a *consistency checking* reasoning task:

`performedIn(CURR_ACT,CURR_LOC_1);`

`isABoxConsistent().`

The above statements are used to verify if activity `BrushingTeeth` can be performed in location `RestRoom`. In this case the consistency check succeeds, since the declared constraints on the execution of `BrushingTeeth` (i.e., the presence of a sink) are satisfied by the considered location. The same statements, substituting `CURR_LOC_1` with `CURR_LOC_2` verify if activity `BrushingTeeth` can be performed in location `LivingRoom`. In this case the consistency check does not succeed, since the definition of `LivingRoom` states that no `WaterFixture` is present in that location. As a consequence, since `Sink` has been defined as a subclass of `WaterFixture`, the ontological reasoner infers that no sink is present in `LivingRoom`, thus violating the constraints for the execution of activity `BrushingTeeth`.

4.2.3 DPA algorithm

Figure 5 shows the algorithm for the Derivation of Possible Activities (named *DPA* algorithm). This algorithm is executed by the offline ontological reasoning module. The algorithm takes as input an empty ABox and the terminological part of the ontology (called TBox) that describes classes and their properties. The output of the algorithm is a matrix M whose rows correspond to symbolic locations in the TBox, columns correspond to activities in the TBox, and $M_{i,j}$ equals to 1 if activity corresponding to column j is a possible activity in location corresponding to row i according to the TBox; $M_{i,j}$ equals to 0 otherwise.

As a first step (line 2) the terminological part of the ontology is classified to compute the hierarchy of the concepts of the TBox. Then for each pair $\langle l_i, a_j \rangle$, where l_i is a symbolic location and a_j is an activity in TBox, the algorithm creates three assertions s_1, s_2 and s_3(lines 5 to 7) to state that activity a_j is performed in location l_i, and adds them to the ABox (line 8). Then (line 9), the ABox is checked for consistency, and $M_{i,j}$ is set with the result of the test (1 if the check succeeds, 0 otherwise). Finally (line 10) assertions s_1, s_2 and s_3 are retracted from the ABox in order to remove the possible inconsistency that would affect the result of future consistency checks.

***DPA* Algorithm**
Input: TBox is the terminological part of the ontology (i.e., containing classes descriptions). In particular, *Activities* is the set of descendants of *Activity*, and *Locations* is the set of descendants of *SymbolicLocation*; ABox is the assertional part of the ontology (i.e., containing individuals and their relationships).
Output: the matrix of potential activities M

```
 1: DPA(TBox,ABox):
 2: ClassifyTBox(TBox)
 3: for all $a_j \in$ Activities do
 4:     for all $l_i \in$ Locations do
 5:         $s_1$ := Assertion("$a_j$(curr_act)")
 6:         $s_2$ := Assertion("$l_i$(curr_loc)")
 7:         $s_3$ := Assertion("performedIn(curr_act, curr_loc)")
 8:         ABox.addAssertions($s_1, s_2, s_3$)
 9:         $M_{i,j}$ := ABox.isConsistent()
10:         ABox.retractAssertions($s_1, s_2, s_3$)
11:     end for
12: end for
13: Return(M)
```

Fig. 5 Algorithm for the derivation of potential activities (DPA).

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Garden	0	0	0	1	1	1	1	0	0	0
HospitalBuilding	1	0	0	0	0	1	0	1	1	1
Kitchen	1	0	0	0	0	1	0	0	0	1
Laboratory	0	0	0	0	0	1	0	0	0	1
LivingRoom	0	0	0	0	0	1	0	0	0	0
Meadow	0	0	0	1	1	1	1	0	0	0
RestRoom	1	0	0	0	0	1	0	0	0	0
UrbanArea	0	0	0	1	1	1	1	1	1	0
Wood	0	1	1	1	1	1	1	0	0	0

Columns: 1=**brushingTeeth**; 2=**hikingUp**; 3=**hikingDown**; 4=**ridingBycicle**; 5=**jogging**; 6=**standingStill**; 7=**strolling**; 8=**walkingDownstairs**; 9=**walkingUpstairs**; 10=**writingOnBlackboard**

Table 2 Part of the M matrix of potential activities

An example of the output of the DPA algorithm with a subset of locations and activities modeled by COSAR-ONT is given in Table 2.

5 Hybrid statistical-ontological reasoning

In this section we illustrate the hybrid technique to couple statistical and ontological reasoning.

5.1 Ontological refinement of statistical predictions

We illustrate our technique by means of an example.

Example 5 Suppose that user Alice is taking a stroll on a path that goes across the wood near home wearing the sensor equipment of the monitoring system. As explained before, the system (deployed on her mobile phone) continuously keeps track of her current activity, as well as of her current symbolic location (that in this case is `Wood`). The system also knows the matrix M that was calculated offline by the DPA algorithm.

Considering a single activity instance i and the statistical model of m different activities $a_1, \ldots, a_m$, the statistical classifier of the system returns a m-length *confidence vector* $\overrightarrow{s}_i$ in which the j^{th} element $\overrightarrow{s}_i^{(j)}$ corresponds to activity a_j and its value corresponds to the *confidence* of the classifier regarding the association of i to a_j, such that:

$$0 \leq \overrightarrow{s}_i^{(j)} \leq 1,$$

and:

$$\sum_{j=1}^{m} (\overrightarrow{s}^{(j)}) = 1.$$

For instance, suppose that the considered activities are those shown in Table 2 (the j^{th} column of the table corresponds to activity a_j), and that:

$$\overrightarrow{s}_i = \langle 0, 0, 0.16, 0, 0, 0, 0.39, 0.45, 0, 0 \rangle.$$

In this case, the maximum confidence value (0.45) corresponds to activity `WalkingDownstairs`, followed by `Strolling` (0.39) and `hikingDown` (0.16). The confidence value corresponding to the other seven activities is 0. Hence, considering the statistical prediction alone, the classifier would erroneously conclude that user Alice is walking downstairs.

However, looking at matrix M one can note that `WalkingDownstairs` is not a feasible activity in the current location of Alice. The rationale of the COSAR technique is to discard those elements of $\overrightarrow{s}_i$ that correspond to unfeasible activities according to M, and to choose the activity having maximum confidence among the remaining elements (or one such activity at random if the maximum confidence corresponds to more than one activity). In this case, the COSAR technique consists in discarding activities `BrushingTeeth`, `WalkingDownstairs`, `WalkingUpstairs` and `WritingOnBlackboard`, and in choosing activity `Strolling`, since it is the one that corresponds to the maximum confidence among the remaining activities. Hence, in this case the COSAR technique correctly recognizes Alice's activity.

5.2 Handling location uncertainty

Every localization technology is characterized by a certain level of inaccuracy. As a consequence, the mapping of a physical location reading to a symbolic location is prone to uncertainty. For instance, if the physical location is retrieved from a GSM cell identification system, the area including the user may correspond to different symbolic locations, such as a `HomeBuilding`, a `HospitalBuilding` and a `Park`.

Uncertainty in location is taken into account by our system. In particular, if the user's physical location corresponds to n possible symbolic locations $l_1, \ldots, l_n$, the possible activities that can be performed by the user are calculated as those that can be performed in at least one location belonging to $\{l_1, \ldots, l_n\}$.

Example 6 Suppose that Alice forgot her GPS receiver at home. Consequently she relies on a GSM cell identification service, which provides coarse-grained location information. In particular, the service localizes Alice within an area that includes both a `Wood` and a `UrbanArea`. Hence, our system calculates the set of Alice's possible activities as the union of the set of activities that can be performed in woods and the set of activities that can be performed in urban areas. Considering matrix M derived by the DPA algorithm and shown in Table 2, possible activities for Alice are those that correspond to columns 2 to 9, included. Therefore, with respect to the scenario depicted in Example 5, in this case `WalkingDownstairs` and `WalkingUpstairs` are possible activities (since urban areas may include steps).

5.3 Recognition of complex activities

As illustrated in Section 2, effective recognition of complex activities like social ones can be hardly achieved by using solely statistical methods. Indeed, symbolic techniques should be used to state the conditions that determine the recognition of a given activity. For instance, possible conditions to recognize the activity *giving a class* are: *"the actor is a teacher, the actor's current location is a classroom, some students are in the classroom, and the actor is writing on a blackboard or using a projector"*.

From this example, it is evident that recognizing this and similar activities relies not only on a symbolic representation of the considered activities, but also on methods to acquire and transform basic observations (e.g., users' presence, simple human activities) into the symbolic representation. Our proposed technique integrates ontological reasoning with the recognition of these basic observations through statistical analysis of sensor data. As explained in Section 3, this technique is executed by the online ontological reasoner in a *best-effort* fashion.

In particular, in order to recognize complex activities, we define them in the terminological part of the ontology, possibly referring to simpler activities that can be recognized through statistical reasoning. For instance, the activity *giving a class* is represented in the ActivO ontology as follows:

$$\begin{aligned}&\texttt{GivingClass} \equiv \texttt{ProfessionalActivity} \sqcap\\ &\quad \forall\, \texttt{hasActor.}\Big(\texttt{Teacher} \sqcap\\ &\qquad \texttt{hasCurrentLocation.}\big(\ \texttt{Classroom} \sqcap\\ &\qquad\quad \geq 2\ \texttt{isInside.Student}\big) \sqcap\\ &\qquad \texttt{hasCurrentActivity.}\big(\ \texttt{WritingOnBlackboard} \sqcup\\ &\qquad\qquad\qquad\qquad \texttt{UsingProjector}\big)\Big).\end{aligned}$$

Activity *business meeting* is represented by the following axiom:

$$\begin{aligned}&\texttt{BusinessMeeting} \sqsubseteq \texttt{SocialActivity} \sqcap\\ &\qquad\qquad \forall\texttt{hasActor.}\big(\texttt{Employee} \sqcap\\ &\exists\,\texttt{hasCurrentLocation.}(\texttt{ConferenceRoom} \sqcap\\ &\qquad\qquad\qquad \texttt{CompanyBuilding})\big),\end{aligned}$$

stating that a social activity in which every actor is an employee whose current location is a company's conference room, is a business meeting.

In order to recognize such activities, basic observations from sensors are mapped into the assertional part of the ontology. For instance, if an indoor positioning system is available, the ontology is kept up-to-date with instances representing people in the smart space; e.g., *"the current location of students Alice and Bob and of teacher Carl is classroom C1"*). Similarly, upon recognition of basic observations through statistical inferencing, the MOBILE DEVICE layer updates the assertional part of the ontology; e.g., *"the current activity of Carl is* writing on a blackboard*"*). When new information is available, a new instance of class *Activity* is created having as actor the current user, and ontological reasoning is activated to derive the most specific class that is instantiated by the user's activity. Continuing the example, if the above conditions are met, the ontological reasoner derives that the most specific activity of Carl is *giving a class*. Then, this information is communicated to the MOBILE DEVICE layer.

5.4 The COSAR module

***COSAR-hist* Algorithm**

Input: τ is the time duration of the activity instances to be classified; n is the number of sensors readings to be acquired during τ; LS is a location server mapping physical locations to symbolic locations; S is a set of sensors; SDF is the sensor data fusion module; M is the matrix of potential activities obtained by the DPA algorithm; *model* is the statistical model of considered activities produced by the pattern recognition module; k is the length of the time window used by the historical algorithm.

Output: $\overrightarrow{C}' = \langle \overline{c}'_1, \overline{c}'_2, \ldots, \overline{c}'_n \rangle$ is the vector of predictions, where prediction $\overline{c}'_i$ refers to the prediction regarding the i-th activity instance.

```
 1: COSAR-hist($\tau, n$, LS, $S$, SDF, $M$, model, $k$):
 2: $\overrightarrow{C}' := \langle\rangle$
 3: $i := 0$
 4: repeat
 5:    $i := i + 1$
 6:    $\overrightarrow{l}_i :=$ LS.getPossibleCurrentLocations()
 7:    $R_i := \{\}$
 8:    for all $s \in S$ do
 9:       $R_i := R_i \cup$ s.takeSamples($\tau, n$)
10:    end for
11:    $f_i :=$ SDF.buildFeatureVector($R_i$)
12:    $\overrightarrow{s}_i :=$ statisticalClassification($f_i$, model)
13:    $\overline{c}_i :=$ COSARPrediction($\overrightarrow{s}_i, \overrightarrow{l}_i, M$)
14:    $\overline{c}'_i :=$ COSAR-histPrediction($\{\overline{c}_i, \overline{c}_{i-1}, \ldots, \overline{c}_{i-k+1}\}$)
15:    $\overrightarrow{C}'$.append($\overline{c}'_i$)
16: until (interruptReceived)
17: Return($\overrightarrow{C}'$)
18:
19: COSARPrediction($\overrightarrow{s}, \overrightarrow{l}, M$):
20: $\overrightarrow{a} := \overrightarrow{l} \times M$
21: COSAR-p $:= j$ s.t. $s^{(j)} = \max\{s^{(h)} | a^{(h)} \neq 0\}$
22: Return(COSAR-p)
23:
24: COSAR-histPrediction($\{\overline{c}_i, \overline{c}_{i-1}, \ldots, \overline{c}_{i-k+1}\}$):
25: $(A, m) :=$ multiset($\{\overline{c}_i, \overline{c}_{i-1}, \ldots, \overline{c}_{i-k+1}\}$)
26: COSARhist-p $:= j$ s.t. $m(j) = \max\{m(i)\}$, $i \in A$
27: Return(COSARhist-p)
```

Fig. 6 Historical algorithm for combined ontological-statistical activity recognition (COSAR-hist algorithm).

This module is in charge of executing the historical variant of the algorithm for combined ontological-statistical activity recognition shown in Figure 6. At first, the algorithm initializes the vector of predictions $\overrightarrow{C}'$ (line 2). Then, the process of actual activity recognition starts, and continues until an interrupt is received (lines 4 to 16). For each activity instance to be recognized, the LOCATION SERVER is queried to obtain the symbolic location corresponding to the current physical location of the user (line 6). Note that more than one symbolic location can correspond to the user's physical location; for instance, if the location server provides location information at a coarse grain. Then, raw data are retrieved from sensors (lines 8 to 10) and a feature vector f_i is built by the SENSOR DATA FUSION module (line 11). The feature vector is used to classify the corresponding activity instance according to the statistical model provided offline by the PATTERN RECOGNITION module, obtaining a confidence vector $\overrightarrow{s}_i$ (line 12). According to $\overrightarrow{s}_i$, to the possible symbolic locations $\overrightarrow{l}_i$, and to the matrix M obtained by the DPA algorithm, the combined ontological-statistical prediction $\overline{c}_i$ is calculated by the *COSARPrediction* procedure (lines 19 to 22). As explained before, $\overline{c}_i$ is the potential activity having highest confidence in $\overrightarrow{s}_i$. Finally, the historical variant is applied to obtain the COSAR-hist prediction $\overline{c}'_i$ (lines 24 to 27) considering the given time window. As explained in Section 4.1, $\overline{c}'_i$ is calculated considering the predictions $\{\overline{c}_i, \overline{c}_{i-1}, \ldots, \overline{c}_{i-k}\}$ of the statistical-hist classifier (in line 26, with $m(j)$ we denote the multiplicity of element j in the multiset of predictions). Then, prediction $\overline{c}'_i$ is added to the vector of predictions $\overrightarrow{C}'$ (line 15).

6 Implementation

In order to experiment our technique we have developed a prototype implementation of the COSAR system for commercially available sensor systems and mobile devices. In our prototype implementation, we reproduce the situation in which data are acquired from an accelerometer embedded in a fitness watch, an accelerometer embedded in a mobile phone, and a localization technology providing the user's current symbolic location. Such data are sufficient to recognize different physical activities, as well as some activities of daily living. We have also developed a prototype application that takes advantage of COSAR to automatically adjust the input/output modalities of a smart phone based on the current activity; in Figure 7, the user interface shows the current recognized activity (*standing still*).

6.1 Sensor layer

The SENSOR layer of our architecture has been implemented on a smart phone running the Android[2] platform, and on *Small Programmable Object Technology (SPOT[3])* sensors by Sun® Microsystems.

The Android device we used is a HTC Magic device, having a 528MHz processor, 288MB RAM/512MB ROM memory, integrated 3-axis accelerometer and digital compass. We developed a simple Java application to provide the upper layer with data about accelerations of the hip (we assume that the user holds the device in her pocket).

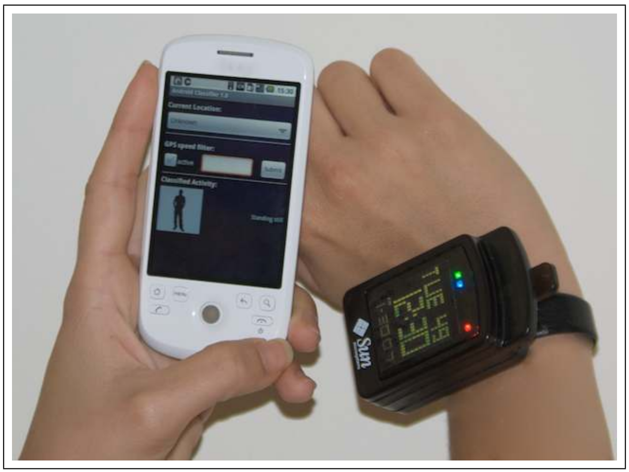


Fig. 7 The Android smart phone and Sun SPOT sensor used for the experiments

Sun SPOTs are sensor devices programmable in Java Micro Edition; they are equipped with a 180 MHz 32 bit processor, 512K RAM/4M Flash memory, and IEEE 802.15.4 radio with integrated antenna. They mount a 3-axis accelerometer, and sensors for light intensity and temperature. SPOT sensors include a prototype of fitness watch (shown in Figure 7) that provides acceleration of the right wrist through the 3-axis accelerometer of the SPOT, and a SPOT widget to provide the current symbolic location.

The sensor layer is in charge of communicating sensor data to the COSAR module; it does not perform any intensive computation such as feature calculation or reasoning. In the current implementation, accelerometers take samples at 16Hz frequency, and communicate them to the MOBILE DEVICE layer. Moreover, upon changes of the user's symbolic location, the SPOT widget communicates the new location to the upper layer.

6.2 Mobile device layer

The MOBILE DEVICE layer has been implemented in Java for the Android platform. It is in charge of building feature vectors based on sensor data, and to execute the *COSAR-hist* algorithm shown in Figure 6 to recognize simple activities at run time. In particular, the statistical classifier that has been adopted is the multiclass logistic regression algorithm, which has been chosen for its efficiency and good recognition rates (see Section 7.2 for more details).

[2] http://www.android.com

[3] http://www.sunspotworld.com

6.3 Infrastructure layer

The main components of the infrastructure layer are the module for pattern recognition and the ontological reasoner.

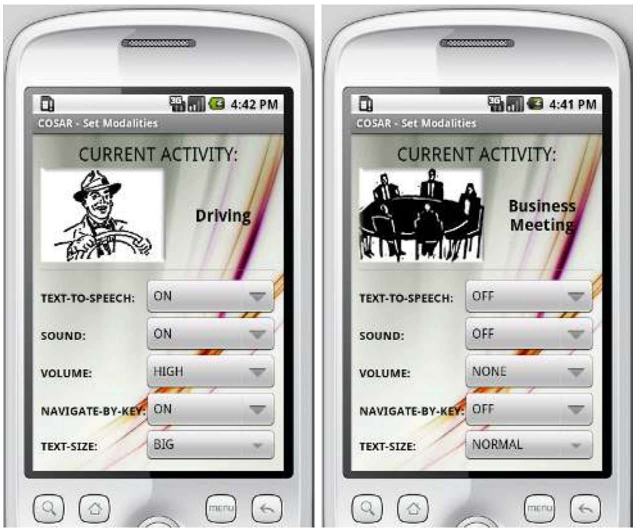
The pattern recognition module is in charge of deriving the statistical model of considered activities based on acquired training data. To this aim, we have used *Weka*[4], a Java-based toolkit that provides APIs for implementing several machine learning algorithms. We have performed extensive experiments (reported in Section 7.2) with different algorithms, and finally we have chosen multiclass logistic regression to derive the model of activities. This model is communicated offline to the MOBILE DEVICE layer.

The AttivO ontology was developed using *Protégé*[5], a graphical tool for ontology development that simplifies design and testing. Ontological reasoning is performed on a two-processor Xeon 2.4GHz workstation with 1.5GB of RAM, using a Linux operating system, the *RacerPro*[6] reasoner, and its APIs for the Java programming language. In particular, the DPA algorithm (Figure 5) has been developed in Java, and the matrix of potential activities is communicated offline to the MOBILE DEVICE layer. Similarly, we developed the module for recognition of complex activities in Java, applying the optimizations for online ontological reasoning presented in [1].

Since SPOT sensors lacked a Bluetooth interface we could not establish a direct connection between the MOBILE DEVICE and the SENSOR layer. For this reason, the INFRASTRUCTURE layer includes a simple Java application to forward packets from the SPOTs to the Android device, and vice-versa. This application communicates with remote SPOTs through a dedicated Sun SPOT basestation using the IEEE 802.15.4 wireless network protocol. Communication between the INFRASTRUCTURE and MOBILE DEVICE layer relies on WiFi and Bluetooth interfaces.

6.4 Services layer

Even if the development of activity-aware applications is out of the scope of this paper, we mention that, in order to evaluate the effectiveness of our solution in real-life situations, at the time of writing we are developing some prototype services for different scenarios, including well-being, context-aware retrieval of georeferenced resources, and adapted interaction. One of them, developed for the Android platform and shown in Figure 8, is in charge of automatically setting the input and output modalities of the smart phone based on user's preferences and current activity. Decisions about the chosen modality can be applied both to local applications (for instance, the phone is muted if the user is in a business meeting), and to Web applications, e.g., using the techniques proposed in [20]. For instance, if the user is driving, text-to-speech can be automatically activated, and image size, color, contrast and lightness can be modified by a transcoding intermediary to improve readability.



COSAR - Set Modalities

CURRENT ACTIVITY:

Driving

TEXT-TO-SPEECH: ON

SOUND: ON

VOLUME: HIGH

NAVIGATE-BY-KEY: ON

TEXT-SIZE: BIG

COSAR - Set Modalities

CURRENT ACTIVITY:

Business Meeting

TEXT-TO-SPEECH: OFF

SOUND: OFF

VOLUME: NONE

NAVIGATE-BY-KEY: OFF

TEXT-SIZE: NORMAL

(a) Current activity: driving (b) Current activity: business meeting

Fig. 8 *Android* implementation

7 Experimental evaluation

In order to validate our solution we performed an extensive experimental evaluation comparing our technique with a purely statistical one. We point out that the symbolic location is used as a feature only in the experiments performed with the purely statistical technique (named *statistical* and *statistical-hist* in the following). In the experiments with the COSAR technique (named *COSAR* and *COSAR-hist*) location is not used as a feature by the statistical classifier; instead, it is used by the ontological module only.

7.1 Experimental setup

The experiments concerned the recognition of 10 different activities performed both indoor and outdoor by 6 volunteers (3 men and 3 women, ages ranging from 30 to 60) having different attitude to physical activities. Each

[4] `http://www.cs.waikato.ac.nz/ml/weka/`

[5] `http://protege.stanford.edu/`

[6] `http://www.racer-systems.com/`

(a) Evaluation of statistical classifiers

Classifier	Accuracy
Bayesian Network	72.95%
C4.5 Decision Tree	66.23%
Multiclass Logistic Regression	80.21%
Naive Bayes	68.55%
SVM	71.81%

(b) Overall accuracy

Classifier	Accuracy
statistical	80.21%
statistical-hist	84.72%
COSAR	89.20%
COSAR-hist	93.44%

(c) Error reduction

versus →	statistical	statistical-hist	COSAR
statistical-hist	22.79%		
COSAR	45.43%	29.32%	
COSAR-hist	66.85%	57.07%	39.26%

Table 3 Summary of experimental results

activity was performed by 4 different volunteers for 450 seconds each. While performing activities, volunteers wore one sensor on their left pocket and one sensor on their right wrist to collect accelerometer data, plus a GPS receiver to track their current physical location, which was later mapped to the corresponding symbolic location. Overall, each activity was performed for 30 minutes; hence, the dataset is composed of 5 hours of activity data. The dataset is published on the web site of our project[7] and can be freely used to reproduce the experiments, or as a testbed for evaluating other techniques.

Samples from accelerometers were taken at 16Hz, and the time extent of each activity instance was 1 second; hence, the dataset is composed of $18,000$ activity instances. For each activity instance, accelerometer readings were merged to build a feature vector composed of 148 features, including means, variances, correlations, kurtosis, and other statistical measures.

In order to evaluate recognition rates we performed 4-folds cross validation, dividing the dataset in 4 subsamples such that each subsample contains 450 instances for each activity. Ideally, an out-of-the-box activity recognition system should be able to recognize one person's activities without the need of being trained on that person. Hence, in order to avoid the use of activity data of the same user for both training and testing we ensured that activity instances regarding a given volunteer did not appear in more than one subsample.

7.2 Accuracy

Exp. 1) Statistical classification algorithms: The first set of experiments was only aimed at choosing a statistical classification algorithm to be used in the subsequent experiments. In general, since in many applications activity recognition must be performed on-line, the choice of a classification algorithm should privilege not only good recognition performance, but also very efficient classification procedures. Indeed, in many cases, the activity recognition algorithm must be executed on a resource-constrained mobile device.

In this first experiment we compared classification techniques belonging to different classes of pattern recognition algorithms (i.e., Bayesian approaches, decision trees, probabilistic discriminative models and kernel machines). Experimental results on our data (shown in Table 2(a)) show that, among the considered techniques, Multiclass Logistic Regression with a ridge estimator (MLR), outperform the other techniques, gaining recognition rates higher than 80%. Hence, our choice for the statistical classification algorithm was to use MLR [16], a classification technique belonging to the class of probabilistic discriminative models [4], having the advantage of being particularly computationally efficient at classification time.

Exp. 2) Statistical technique: Table 4 shows the confusion matrix and precision/recall measures for the statistical technique evaluated in the first set of experiments. As expected, when data from accelerometers are used and the symbolic location is used as a feature, many misclassifications occur between activities that involve similar body movements; e.g., instances of *strolling* are often classified as instances of *walking downstairs*; instances of *brushing teeth* are often classified as instances of *writing on blackboard*.

Exp. 3) Statistical-hist technique: We evaluated the historical variant of the statistical classification algorithm by simulating the case in which a user performs each activity for 7.5 minutes before changing activity. With this technique, the accuracy of activity recognition is 84.72% (see Table 2(b)), which results in an error reduction rate of 22.79% with respect to the statistical technique (see Table 2(c)). Observing the confusion matrix and precision/recall measures shown in Table 5, it can be noted that this technique does not significantly reduce the number of misclassifications between activities involving similar movements.

Exp. 4) COSAR technique: The use of the COSAR technique considerably improves the recognition rate with respect to the solely statistical techniques. In particular, the recognition rate of COSAR is 89.2%, which results in an error reduction of 45.43% with respect to

[7] http://everywarelab.dico.unimi.it/palspot

	(a) Confusion matrix										(b) Precision / recall	
classified as →	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**prec.**	**recall**
brushingTeeth	**1336**	4	1	11	8	304	0	33	2	101	95,43%	74,22%
hikingUp	4	**1551**	219	5	14	0	1	1	5	0	76,93%	86,17%
hikingDown	0	382	**1376**	4	3	1	31	2	1	0	80,66%	76,44%
ridingBycicle	1	5	10	**1738**	23	0	0	23	0	0	96,61%	96,56%
jogging	13	3	17	21	**1664**	1	7	73	1	0	71,94%	92,44%
standingStill	32	5	3	0	290	**1254**	17	34	126	39	61,20%	69,67%
strolling	0	0	78	0	304	3	**917**	383	115	0	92,53%	50,94%
walkingDownstairs	0	0	1	0	0	0	2	**1762**	35	0	71,74%	97,89%
walkingUpstairs	0	5	0	4	0	1	16	144	**1629**	1	84,89%	90,50%
writingOnBlackboard	14	61	1	16	7	485	0	1	5	**1210**	89,56%	67,22%

Columns: 1=brushingTeeth; 2=hikingUp; 3=hikingDown; 4=ridingBycicle; 5=jogging; 6=standingStill; 7=strolling; 8=walkingDownstairs; 9=walkingUpstairs; 10=writingOnBlackboard

Table 4 Results for the statistical classifier

	(a) Confusion matrix										(b) Precision / recall	
classified as →	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**prec.**	**recall**
brushingTeeth	**1344**	0	0	8	3	375	0	8	0	62	98.39%	74.67%
hikingUp	3	**1613**	170	0	13	0	1	0	0	0	84.27%	89.61%
hikingDown	8	252	**1525**	1	0	1	13	0	0	0	86.11%	84.72%
ridingBycicle	0	7	5	**1784**	2	0	0	2	0	0	97.65%	99.11%
jogging	4	0	1	18	**1764**	0	0	13	0	0	75.13%	98.00%
standingStill	4	2	0	0	302	**1357**	1	5	97	32	61.57%	75.39%
strolling	0	0	70	0	252	0	**1067**	355	56	0	98.07%	59.28%
walkingDownstairs	0	0	0	0	2	7	0	**1784**	7	0	80.51%	99.11%
walkingUpstairs	0	0	0	0	0	0	0	49	**1751**	0	91.63%	97.28%
writingOnBlackboard	3	40	0	16	10	464	6	0	0	**1261**	93.06%	70.06%

Columns: 1=brushingTeeth; 2=hikingUp; 3=hikingDown; 4=ridingBycicle; 5=jogging; 6=standingStill; 7=strolling; 8=walkingDownstairs; 9=walkingUpstairs; 10=writingOnBlackboard

Table 5 Results for the statistical-hist classifier

	(a) Confusion matrix										(b) Precision / recall	
classified as →	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**prec.**	**recall**
brushingTeeth	**1622**	0	0	0	0	178	0	0	0	0	98,30%	90,11%
hikingUp	0	**1443**	171	19	34	14	119	0	0	0	83,99%	80,17%
hikingDown	0	268	**1284**	22	2	13	211	0	0	0	87,82%	71,33%
ridingBycicle	0	4	7	**1787**	1	1	0	0	0	0	86,87%	99,28%
jogging	0	0	0	134	**1640**	9	6	8	3	0	96,76%	91,11%
standingStill	0	3	0	26	9	**1738**	21	1	2	0	76,06%	96,56%
strolling	0	0	0	69	9	54	**1597**	67	4	0	81,73%	88,72%
walkingDownstairs	4	0	0	0	0	1	0	**1753**	42	0	90,55%	97,39%
walkingUpstairs	24	0	0	0	0	26	0	107	**1643**	0	96,99%	91,28%
writingOnBlackboard	0	0	0	0	0	251	0	0	0	**1549**	100,00%	86,06%

Columns: 1=brushingTeeth; 2=hikingUp; 3=hikingDown; 4=ridingBycicle; 5=jogging; 6=standingStill; 7=strolling; 8=walkingDownstairs; 9=walkingUpstairs; 10=writingOnBlackboard

Table 6 Results for the COSAR classifier

	(a) Confusion matrix										(b) Precision / recall	
classified as →	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**prec.**	**recall**
brushingTeeth	**1716**	0	0	0	0	80	0	0	0	4	99.36%	95.33%
hikingUp	0	**1604**	116	10	16	1	53	0	0	0	87.03%	89.11%
hikingDown	11	235	**1439**	11	0	4	100	0	0	0	92.36%	79.94%
ridingBycicle	0	4	3	**1792**	0	0	1	0	0	0	92.04%	99.56%
jogging	0	0	0	91	**1708**	0	0	1	0	0	98.39%	94.89%
standingStill	0	0	0	17	12	**1748**	23	0	0	0	82.53%	97.11%
strolling	0	0	0	26	0	27	**1698**	40	9	0	90.18%	94.33%
walkingDownstairs	0	0	0	0	0	9	0	**1790**	1	0	96.81%	99.44%
walkingUpstairs	0	0	0	0	0	9	0	18	**1773**	0	99.44%	98.50%
writingOnBlackboard	0	0	0	0	0	240	8	0	0	**1552**	99.74%	86.22%

Columns: 1=brushingTeeth; 2=hikingUp; 3=hikingDown; 4=ridingBycicle; 5=jogging; 6=standingStill; 7=strolling; 8=walkingDownstairs; 9=walkingUpstairs; 10=writingOnBlackboard

Table 7 Results for the COSAR-hist classifier

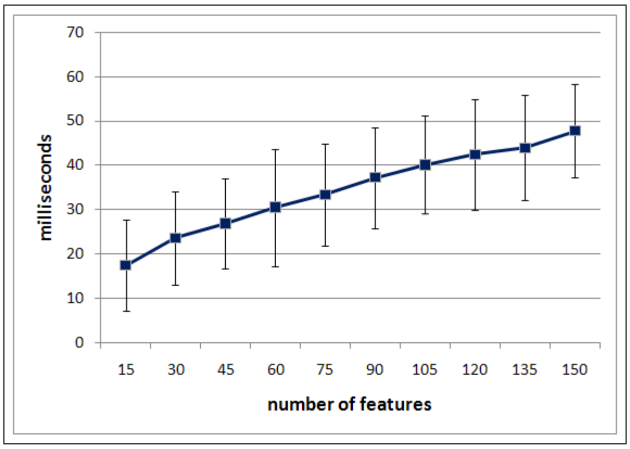


Fig. 9 Computational cost on the mobile device

the statistical technique, and of 29.32% with respect to the statistical-hist technique. Looking at the confusion matrix (Table 6), we note that COSAR avoids many misclassifications between activities characterized by similar body movements but different contexts (e.g., location) in which they are typically performed.

Exp. 5) COSAR-hist technique: Finally, the historical variant of COSAR (evaluated with the same setup as in *Exp. 3*) further improves classification results, gaining a recognition rate of 93.44%, an error reduction of 39.26% with respect to the COSAR technique, and of 66.85% with respect to the statistical technique. Confusion matrix and precision/recall measures are shown in Table 7.

7.3 Computational cost

Consumption of computational resources is a fundamental aspect to be considered, especially when activity recognition is performed using mobile devices. In order to assess the feasibility of our approach, we have conducted experiments regarding the online execution of the *COSAR-hist* algorithm on an Android smart phone, whose capabilities have been reported in Section 6.1.

As explained in Section 3, the MOBILE DEVICE is not in charge of very computationally expensive tasks such as ontological reasoning; however, it is in charge of building feature vectors from sensor data streams, and of performing statistical classification using those vectors. Note that computational cost depends not only on the used algorithm, but also on the size of feature vectors. Indeed, in general the more features are considered, the more expensive is to perform classification. For this reason, we have conducted experiments with different numbers of features; i.e., ranging from 15 to 150. The considered features size is reasonable, as shown in research studying the role of feature selection [11] for activity recognition (see, e.g., [17]). Features were selected from the initial corpus using the well-known *chi-square* [19] method.

With each considered number of features, we have executed the COSAR-hist algorithm on the smart phone for 30 minutes, with samples gathered from the embedded accelerometer at 16Hz, and time extent of activity instances of 1 second (i.e., a classification was performed every one second). For each number of features we have measured the average execution time of COSAR-hist; hence, results are average of 1,800 runs of the COSAR-hist algorithm. Results (also showing standard deviation) are reported in Figure 9. It can be observed that the execution time of the algorithm grows linearly with the number of features. This is due to the fact that execution times are dominated by the statistical learning algorithm MLR, whose complexity is linear with respect to the number of features. Execution times range from less than 20ms when using 15 features to less then 50ms when using 150 features. Hence, we can conclude that our algorithm can be executed on currently available mobile devices with very limited interference on their normal operations.

8 Conclusions and future work

In this paper we illustrated COSAR, a system for automatic activity recognition based on the integration of statistical and ontological reasoning. While the main contribution is the design of the hybrid reasoning algorithm running on the mobile device (including the proposal of a novel variant of statistical methods), our work includes an innovative architecture and its complete implementation. Results from extensive experiments with real data confirm the superiority of our approach with respect to purely statistical methods and the possibility of running the core COSAR system modules on mobile devices.

We are considering several extensions of COSAR. Currently, location is the only data used to enact ontological reasoning, but our technique can be easily extended to consider a wider class of context data. In particular, we plan to extend our technique to consider the temporal characterization of activities (e.g., duration), as well as their temporal relationships (i.e., the probability that a given activity a_i is followed by an activity a_j). To this aim, we should design a temporal extension of our ontology, and investigate the use of a probabilistic framework such as Hidden Markov Models as an evolution of our historical prediction procedure. Another interesting extension would be the design of more expressive activity ontologies using the OWL 2 language instead of OWL-DL, and the substitution of the ontological reasoner used in COSAR with one supporting OWL 2.

Acknowledgments

This work has been partially supported by a grant from Sun® Microsystems. The authors would like to thank the volunteers that collaborated to the collection of data used in our experiments.

References

1. Alessandra Agostini, Claudio Bettini, and Daniele Riboni. Hybrid reasoning in the CARE middleware for context awareness. *International Journal of Web Engineering and Technology*, 5(1):3–23, 2009.
2. Oliver Amft and Gerhard Tröster. Recognition of dietary activity events using on-body sensors. *Artificial Intelligence in Medicine*, 42(2):121–136, 2008.
3. Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003.
4. Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2008. ISBN 0387310738.
5. Oliver Brdiczka, James L. Crowley, and Patrick Reignier. Learning Situation Models for Providing Context-Aware Services. In *Proceedings of HCI 2007*, volume 4555 of *Lecture Notes in Computer Science*, pages 23–32. Springer, 2007.
6. Keng-Hao Chang, Shih-Yen Liu, Hao-Hua Chu, Jane Yung-Jen Hsu, Cheryl Chen, Tung-Yun Lin, Chieh-Yu Chen, and Polly Huang. The Diet-Aware Dining Table: Observing Dietary Behaviors over a Tabletop Surface. In *Proceedings of Pervasive 2006*, volume 3968 of *Lecture Notes in Computer Science*, pages 366–382. Springer, 2006.
7. Harry Chen, Tim Finin, and Anupam Joshi. Semantic Web in the Context Broker Architecture. In *Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom 2004)*, pages 277–286. IEEE Computer Society, 2004.
8. Andrew R. Golding and Neal Lesh. Indoor Navigation Using a Diverse Set of Cheap, Wearable Sensors. In *Proceedings of the Third International Symposium on Wearable Computers (ISWC'99)*, pages 29–36. IEEE Computer Society, 1999.
9. T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang. An ontology-based context model in intelligent environments. In *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, San Diego, California, USA*, January 2004.
10. Tao Gu, Zhanqing Wu, XianPing Tao, Hung Keng Pung, and Jian Lu. 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)*, pages 1–9. IEEE Computer Society, 2009.
11. Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
12. Xin Hong, Chris D. Nugent, Maurice D. Mulvenna, Sally I. McClean, Bryan W. Scotney, and Steven Devlin. Evidential fusion of sensor data for activity recognition in smart homes. *Pervasive and Mobile Computing*, 5(3):236–252, 2009.
13. Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003. ISSN 1570-8268.
14. Tâm Huynh and Bernt Schiele. Towards Less Supervision in Activity Recognition from Wearable Sensors. In *Proceedings of the Tenth IEEE International Symposium on Wearable Computers (ISWC 2006)*, pages 3–10. IEEE, 2006.
15. Nicky Kern, Bernt Schiele, and Albrecht Schmidt. Multi-sensor Activity Context Detection for Wearable Computing. In *Proceedings of the First European Symposium on Ambient Intelligence (EUSAI 2003)*, volume 2875 of *Lecture Notes in Computer Science*, pages 220–232. Springer, 2003.
16. S. le Cessie and J.C. van Houwelingen. Ridge Estimators in Logistic Regression. *Applied Statistics*, 41(1):191–201, 1992.
17. Jonathan Lester, Tanzeem Choudhury, Nicky Kern, Gaetano Borriello, and Blake Hannaford. A Hybrid Discriminative/Generative Approach for Modeling Human Activities. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 766–772. Professional Book Center, 2005.
18. Lin Liao, Dieter Fox, and Henry A. Kautz. Location-Based Activity Recognition using Relational Markov Networks. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 773–778. Pro-

fessional Book Center, 2005.

19. Huan Liu and Rudy Setiono. Chi2: Feature selection and discretization of numeric attributes. In *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, pages 388–391. IEEE Computer Society, 1995.
20. Delfina Malandrino, Francesca Mazzoni, Daniele Riboni, Claudio Bettini, Michele Colajanni, and Vittorio Scarano. MIMOSA: Context-aware Adaptation for Ubiquitous Web Access. *Personal and Ubiquitous Computing*, to appear.
21. Nuria Oliver, Eric Horvitz, and Ashutosh Garg. Layered Representations for Human Activity Recognition. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces (ICMI 2002)*, pages 3–8. IEEE Computer Society, 2002.
22. Linda Pareschi, Daniele Riboni, Alessandra Agostini, and Claudio Bettini. Composition and Generalization of Context Data for Privacy Preservation. In *Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2008), Proceedings of the Workshops*, pages 429–433. IEEE Computer Society, 2008.
23. Daniele Riboni and Claudio Bettini. Context-Aware Activity Recognition through a Combination of Ontological and Statistical Reasoning. In *Proceedings of the 6th International Conference on Ubiquitous Intelligence and Computing (UIC)*, volume 5585 of *Lecture Notes in Computer Science*, pages 39–53. Springer, 2009.
24. Daniele Riboni, Linda Pareschi, and Claudio Bettini. Shadow attacks on users' anonymity in pervasive computing environments. *Pervasive and Mobile Computing*, 4(6):819–835, 2008.
25. M. Stikic, T. Huynh, K. Van Laerhoven, and B. Schiele. ADL Recognition Based on the Combination of RFID and Accelerometer Sensing. In *Proceedings of Pervasive Health 2008*, pages 2237–2242. IEEE Computer Society, 2008.
26. Monica Tentori and Jesús Favela. Activity-Aware Computing for Healthcare. *IEEE Pervasive Computing*, 7(2):51–57, 2008.
27. Shiaokai Wang, William Pentney, Ana-Maria Popescu, Tanzeem Choudhury, and Matthai Philipose. Common Sense Based Joint Training of Human Activity Recognizers. In *Proceedings of IJCAI 2007*, pages 2237–2242, 2007.
28. Zhiwen Yu, Zhiyong Yu, Yusa Ko, Xingshe Zhou, and Yuichi Nakamura. Inferring Human Interactions in Meetings: A Multimodal Approach. In *Proceedings of the 6th International Conference on Ubiquitous Intelligence and Computing (UIC)*, volume 5585 of *Lecture Notes in Computer Science*, pages 14–24. Springer, 2009.