CAVIAR: Context-driven Active and Incremental Activity Recognition

Claudio Bettini, Gabriele Civitarese*, Riccardo Presotto

EveryWare Lab, Dept. of Computer Science
Università degli Studi di Milano, Via Celoria 18, Milan, Italy

Abstract

Activity recognition on mobile device sensor data has been an active research area in mobile and pervasive computing for several years. While the majority of the proposed techniques are based on supervised learning, semi-supervised approaches are being considered to reduce the size of the training set required to initialize the model. These approaches usually apply self-training or active learning to incrementally refine the model, but their effectiveness seems to be limited to a restricted set of physical activities. We claim that the context which surrounds the user (e.g., time, location, proximity to transportation routes) combined with common knowledge about the relationship between context and human activities could be effective in significantly increasing the set of recognized activities including those that are difficult to discriminate only considering inertial sensors, and the highly context-dependent ones. In this paper, we propose CAVIAR, a novel hybrid semi-supervised and knowledge-based system for real-time activity recognition. Our method applies semantic reasoning on context-data to refine the predictions of an incremental classifier. The recognition model is continuously updated using active learning. Results on a real dataset obtained from 26 subjects show the effectiveness of our approach in increasing the recognition rate, extending the number of recognizable activities and, most importantly, reducing the number of queries triggered by active learning. In order to evaluate the impact of context reasoning, we also compare CAVIAR with a purely statistical version, considering features computed on context-data as part of the machine learning process.

Keywords: activity recognition, mobile computing, semi-supervised learning, context-awareness, hybrid reasoning

1. Introduction

The rapid evolution of sensor technology and mobile computing in the last decades opened the way to a new generation of intelligent context-aware services relying on the ability to automatically detect our daily activities [1]. Some of those services have real-time constraints, thus requiring activity recognition methods capable of balancing accuracy, fast reaction time, and resource efficiency [2]. The majority of activity recognition algorithms in the literature rely on supervised machine learning to infer the most likely performed activities by analyzing inertial sensors data [3]. One of the major drawbacks of those solutions is the cost of collecting the amount of labeled data required to reach a high recognition rate. Moreover, standard classifiers are trained once with available data, and the recognition model cannot evolve. In order to overcome these issues, semi-supervised and incremental approaches for activity recognition have been proposed [4]. Those methods only require a small

*Corresponding author

Email addresses: `claudio.bettini@unimi.it` (Claudio Bettini), `gabriele.civitarese@unimi.it` (Gabriele Civitarese), `riccardo.presotto@unimi.it` (Riccardo Presotto)

amount of training data to initialize the recognition model, while techniques like co-learning, self-learning or active learning are used to assign labels to the sensor data acquired while the system is deployed [5, 6, 7, 8].

While the majority of semi-supervised methods showed to be effective on classifying a limited set of physical activities (e.g., walking, running, biking, etc.), their effectiveness on more complex and context-dependent activities is still unclear. Moreover, discriminating those activities which have similar motion patterns is still problematic. For instance, activities like *walking* and *taking the stairs*, or *standing* and *taking the elevator* are easily confused between them by purely statistical methods based on inertial sensors.

The context which surrounds the user could be valuable information to mitigate these issues [9, 10]. Indeed, a rich description of the user's context (e.g., semantic location, weather, traffic condition, speed, etc.) has the potential to enable the recognition of a wide set of activities which are a) highly dependent on the current context and b) difficult to recognize only considering inertial sensors. However, semi-supervised approaches rely on a small set of labeled data that may not be representative of the large number of possible context conditions in which activities can be performed. For this reason, directly using context-data as additional features in the machine learning process may not be as effective as expected.

In this paper, we consider this problem and we propose CAVIAR, a *Context-aware ActiVe and Incremental Activity Recognition* system which combines semi-supervised learning and semantic context-aware reasoning. An online machine learning classifier is in charge of inferring from inertial sensor data a candidate probability distribution over the possible activities. A knowledge-based reasoning engine is then used to exclude the activities that are highly unlikely considering context-data. The system provides as output the most likely activity from the resulting context-refined probability distribution.

Following the semi-supervised approach, when CAVIAR is not sufficiently confident about the current activity despite the context-based refinement, it starts an active learning process, asking the user about the activity being performed and using the answer to provide a new labeled sample to the incremental classifier.

No public datasets exist with sufficiently rich context information to properly evaluate CAVIAR. Hence, we conducted a new dataset acquisition campaign involving 26 subjects. Results on this dataset show that context-refinement is effective in a) improving the recognition rate, b) expanding the set of recognizable activities, and c) triggering a significantly lower number of queries.

The contributions of our work are the following:

- We propose CAVIAR, a novel activity recognition approach that combines context-aware reasoning with semi-supervised learning.
- We performed an extensive evaluation of CAVIAR on a real annotated dataset acquired from 26 subjects while performing 13 different activities. The results show the crucial role of context-data and structured knowledge in improving semi-supervised activity recognition.
- We show that CAVIAR, by using knowledge-based reasoning on context-data, not only reaches higher recognition rates but also triggers a significantly lower number of queries for active learning compared to using context-data as additional features in the machine learning process.

The rest of the paper is organised as follows. Section 2 discusses related work. Section 3 describes the overall architecture of CAVIAR. Section 4 presents the CAVIAR method in details. Section 5 reports the experimental results. Section 6 discusses strengths and limitations of CAVIAR. Finally, Section 7 concludes the paper.

2. Related work

The recognition of physical activities using commonly available mobile devices (e.g., smartphones and smartwatches) is a widely explored research area [1, 11]. The majority of approaches in the literature rely on supervised methods to infer activities from inertial sensors data [3, 12, 13, 14, 15]. While these methods allow reaching high recognition rates, the acquisition of a wide labeled dataset of activities is costly and often unfeasible [16].

In order to overcome these issues, a few research works proposed unsupervised learning techniques [17, 18, 19]. These methods aim to find activity patterns from unlabeled data with data mining techniques. However, those approaches require a large pool of data to discover significant patterns. Moreover, a certain amount of labeled data is still required to reliably associate each discovered cluster with its corresponding activity class.

In order to combine the strengths of both supervised and unsupervised approaches, semi-supervised learning methods for activity recognition have been proposed [4, 8, 20, 7]. Those techniques use small labeled training sets to initialize the model, which is continuously enhanced using unlabeled data. In the literature, the semi-supervised strategies used for activity recognition are self-learning, co-learning, and active learning. Self-learning methods exploit the starting training set to classify unlabeled data [7], and the most confident predictions are used to update the classifier. Co-learning approaches rely on multiple classifiers trained on different views of the training set, and on a voting mechanism to classify each unlabeled feature vector [21, 20]. Based on their confidence, the resulting predictions may be used to update the recognition model of each classifier. Differently from self-learning and co-learning, active learning requires explicit feedback from the users to obtain labels for the most informative data (i.e., when the classifier is uncertain about the performed activity) [22, 23, 6, 5]. Active learning proved to be particularly effective for semi-supervised activity recognition. However, for the sake of usability, the number of triggered queries should be as small as possible.

Existing semi-supervised activity recognition methods in mobile computing mainly recognize a restricted number of physical activities using inertial sensors data [23, 6, 21, 24]. Differently from those methods, we consider the context which surrounds the user to refine the statistical predictions and to continuously update the recognition model through active learning. Thanks to this approach, CAVIAR significantly extends the set of recognizable activities and, at the same time, better discriminates those activities which have similar movement patterns.

Even if context reasoning for activity recognition has been mainly investigated for smart-home environments [25] and computer vision based systems [26], its application to mobile computing applications is not completely new [27]. For instance, in [28] the authors propose a supervised method that considers features like the current user's position (e.g., semantic location) and personal attributes (e.g., age, gender, profession). However, the activities considered in this work are strictly related to semantic places (e.g., their model predicts *praying* if the user is in a church, or *shopping* if the user is in some kind of shop, etc.).

The combination of machine learning and context-aware ontological reasoning for activity recognition was firstly explored in [10]. In that work, the output of a statistical classifier is refined considering the user's semantic location. In [29] rich contextual information is used to improve activity recognition with a multi-layer approach. Differently from those methods, our system is semi-supervised and it only requires a small set of labeled data to be initialized. Moreover, CAVIAR takes advantage of context-aware reasoning to continuously update the activity recognition model with active learning.

3. CAVIAR system overview

The general architecture of CAVIAR is depicted in Figure 1. The user's mobile devices continuously collect data from different sources. While inertial sensors (e.g., accelerometer, magnetometer, and gyroscope) stream data about the user's physical movements, CAVIAR gathers data about the user's context by combining data from the devices (e.g., the geographical position) with publicly available web services (e.g., local weather service). It is important to note that "*context*" is a very broad term which in the literature is used to define the user's situation at different levels of abstractions [30]. In this paper, with context data we mainly indicate the information about the environment which surrounds the user. Examples of such context data are the user's current semantic location, the fact that it is indoor or outdoors, his/her proximity to transportation routes, the current weather, temperature, and temporal properties like the time of the day, and the day of the week, etc.

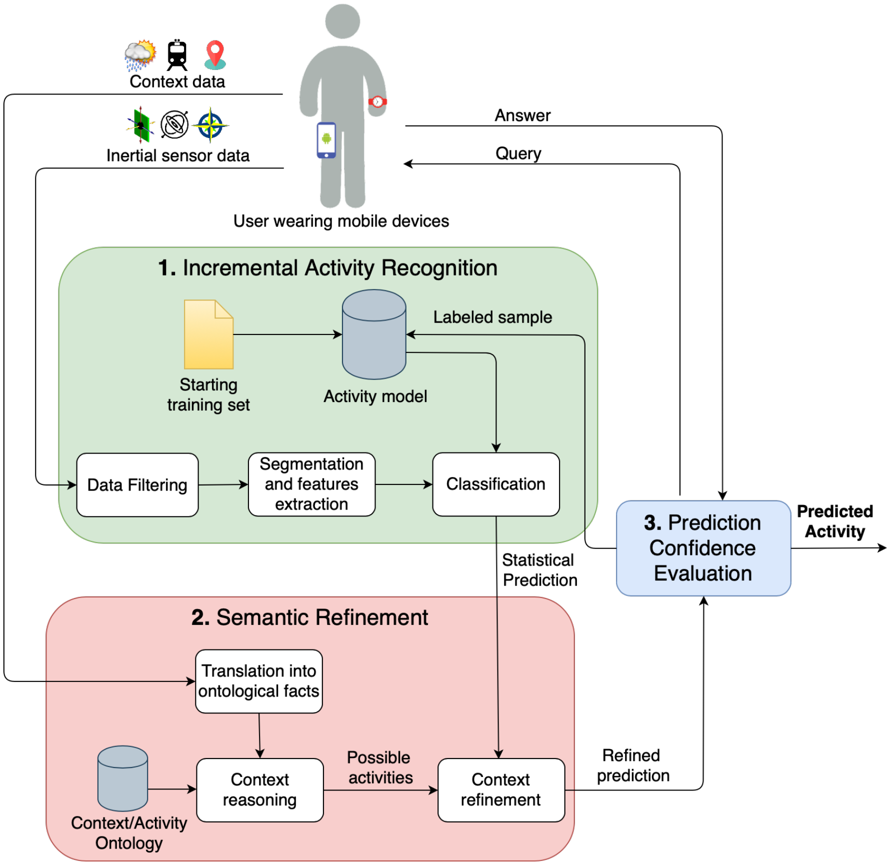


Figure 1: Overall architecture of our system

Our hybrid semi-supervised and context-aware activity recognition algorithm is divided into three main steps. First, the stream of raw inertial sensors readings is processed by the INCREMENTAL ACTIVITY RECOGNITION module. This module first applies pre-processing methods like data filtering, segmentation, and feature extraction. Then, a semi-supervised classifier associates to each feature vector a probability distribution over the possible activities. The activity model is initialized during an offline phase with a small labeled training set.

The INCREMENTAL ACTIVITY RECOGNITION module does not take into account context data. The main motivation is that semi-supervised methods rely on a rather small set of labeled data, while activities can be performed in a wide variety of different contexts. While pure data-driven techniques can discriminate different motion patterns even with a few labeled samples, learning their correlations when they compose an activity in very different context conditions may be problematic. This is especially true when considering a wide set of activities. Nonetheless, common-sense knowledge can be used to model the relationships between activities and contexts (e.g., it is unlikely that a user is using an elevator while she is in the city park). Hence, in the second step of our algorithm, the Semantic Refinement module applies knowledge-based reasoning to context data to exclude from the semi-supervised prediction those activities which are not consistent with the current context. In particular, raw context data is first pre-processed and translated into high-level facts. An ontology that models the relationships between activities and contexts is then used to derive from context data which of the candidate activities are *context-consistent*. The output of the Semantic Refinement module is a probability distribution over the *context-consistent* activities, which we call *refined prediction*.

The third and last step of CAVIAR consists of using the refined prediction to update the activity model. In particular, the Prediction Confidence Evaluation module evaluates the system's confidence on the refined prediction. If the confidence is below a certain threshold, a query is triggered to the user to obtain the ground truth about the current activity. This feedback is then used to provide a new labeled data sample to the activity model.

Note that the system is centralized and the feedback by each user in the active learning phase contributes to the refinement of the shared activity model.

4. Methodology

In this section, we describe in detail how the different modules of CAVIAR introduced in Section 3 actually work.

4.1. Incremental activity recognition

The Incremental Activity Recognition module relies on an online semi-supervised classifier to derive a candidate set of activities performed by the user. The stream of inertial sensor data is continuously pre-processed and segmented to extract feature vectors. As soon as a feature vector is generated by CAVIAR, it is provided to the classifier to derive the probability distribution over all the possible activities. Note that the activity model is first initialized during an offline phase with a small amount of labeled data.

4.1.1. Segmentation, feature extraction and classification

In the following, we describe the pre-processing steps that CAVIAR applies to inertial sensors data. Since a user may carry multiple mobile devices (e.g., a smartphone and a smartwatch), it is first necessary to temporally align their raw sensor data streams. In our experimental setup, we considered for each device the data streams from the accelerometer, magnetometer, and gyroscope. CAVIAR applies to each stream a median filter to reduce the intrinsic noise of the signal. Then, CAVIAR segments the streams of aligned sensor data. Each segment is defined as the set of inertial sensor data acquired during a specific time window of n seconds. Each segment starts the next second with respect to the end of the previous segment, hence segments are contiguous and non-overlapping.

The length n is the same for all segments, and it should be chosen carefully according to the complexity of the considered activities to balance the trade-off between accuracy and reaction time. In our experimental setup, we studied the existing literature to choose a reasonable fixed value for n [31]. Since the target activities of CAVIAR are both simple (e.g., standing) and complex (e.g., driving) we could not choose a too short window size. Hence, to guarantee a reasonable trade-off between accuracy and reaction time, CAVIAR uses $n = 4$.

From each segment, CAVIAR extracts a wide set of statistical features that are well-known in the activity recognition literature [1]. In particular, for each axis of each inertial sensor, we extract: *average*, *variance*, *standard deviation*, *median*, *mean squared error*, *kurtosis*, *symmetry*, *zero-crossing rate*, *number of peaks*, *energy* and *difference between maximum and minimum*. Finally, for each inertial sensor we compute the *Pearson correlation* for each combination of its axes and the *magnitude* on all of its axes. Hence, given k 3-axis inertial sensors equipped in the user's mobile devices, we compute $k \times 37$ features. We also apply standardization to each feature to further improve the recognition rate [32].

Example 4.1. *Consider a user which carries a smartphone and a smartwatch, both equipped with 3-axis accelerometer, gyroscope, and magnetometer. Since the overall number of inertial sensors is* 6*, CAVIAR would compute, for each segment,* $6 \times 37 = 222$ *features.*

For each feature vector fv computed from a segment s, the incremental classifier h outputs a probability distribution over the set of considered activities $\mathbf{A} = \{A_1, A_2, \ldots, A_m\}$:

$$h(fv) = \langle p_1, p_2, \ldots, p_m \rangle$$

where $0 \leq p_i \leq 1$ is the probability $P(A_i|s)$ that the segment s was generated by the activity A_i, with $\sum_i^m p_i = 1$, and $m = |\mathbf{A}|$. The probability distribution $h(fv)$ is forwarded to our Semantic Refinement module which will refine it based on context data.

4.1.2. Activity model bootstrap

A crucial aspect of our semi-supervised framework is the activity model initialization. Indeed, without a proper bootstrap mechanism, the semi-supervised model would have to discover each activity "on-the-fly", with a negative impact on the recognition rate. Hence, we initialize the semi-supervised model by acquiring t seconds of labeled data for each activity to obtain a balanced labeled dataset. While the value of t should be as small as possible to minimize the effort in the labeled set collection, this value has a high impact both on the recognition accuracy and on the number of queries triggered to the users. In our experimental setup, we consider $t = 60$ (i.e., one minute) and hence, considering the length of each segment, CAVIAR initializes the activity model using 15 labeled feature vectors for each activity. Based on our experiments and the range of considered activities, we believe that this small value should be sufficient in general for CAVIAR to initialize an activity model that can rapidly evolve thanks to active learning.

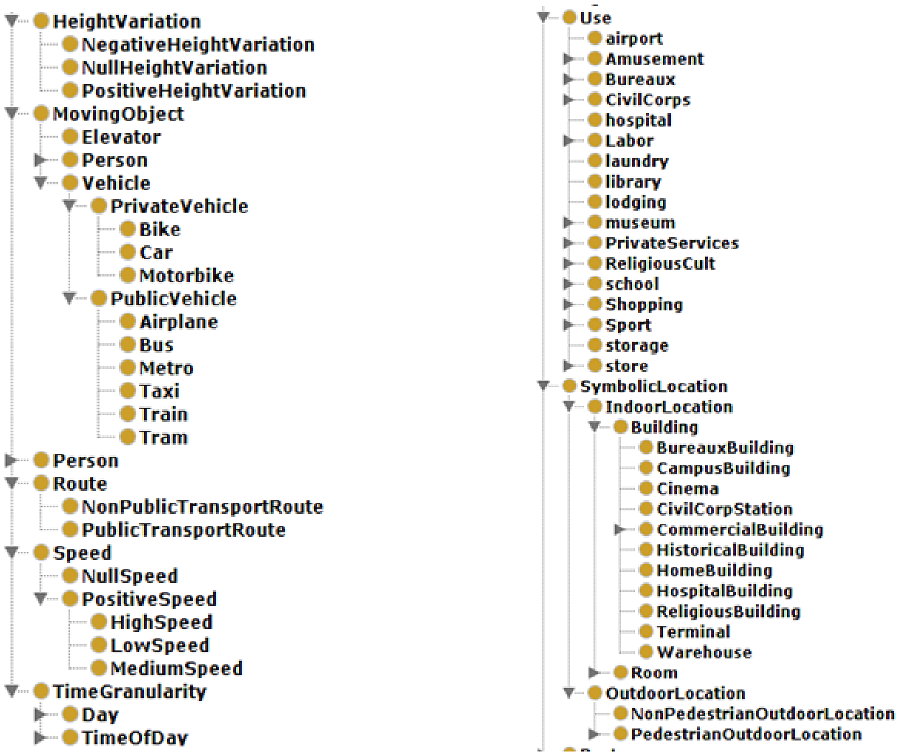
4.2. Semantic Refinement

The Semantic Refinement module is in charge of refining the prediction $h(fv)$ obtained by the Incremental Activity Recognition through the analysis of the context which surrounds the user. In order to achieve this task, this module relies on an ontology that models the relationships between context and activities. Intuitively, ontological reasoning is applied to exclude from the statistical prediction those activities which are unlikely to be performed considering the current context. In the following, we describe in detail our context-aware semantic reasoning mechanism.

4.2.1. Activity and Context ontology

Our ontology is an extension of the *ActivO* ontology [10] that defines a wide set of activities, semantic locations, artifacts (e.g., used by the user or part of the semantic locations), user's postures, time granularities (e.g., day of the week, time of the day) and environmental information (e.g., temperature and light conditions).

Details aboutActivO's implementation can be found in [10]. We took advantage of the Protégé tool [1] to extend *ActivO* with several new activities, contextual data and their relationships. An example of those entities is shown in Figure 2. Our ontology considers several sources of context data: *user's semantic place*, *user's recent route*, *weather conditions*, *proximity to public transportation stops and routes*, *surrounding traffic condition*, *user's height variations*, *user's speed*, *surrounding light*, *environment's noise level* and *temporal context (e.g., time of the day, day of the week, month, . . .)*. Figure 2a shows a portion of those context data modelled in our ontology, while Figure 2b focuses on the set of considered semantic locations, including the ones classified by Google Places API [2]. It is important to note that we distinguish symbolic locations and their characteristics from their use. This allows us to better model activities related to symbolic locations.



(a) An excerpt of context hierarchy

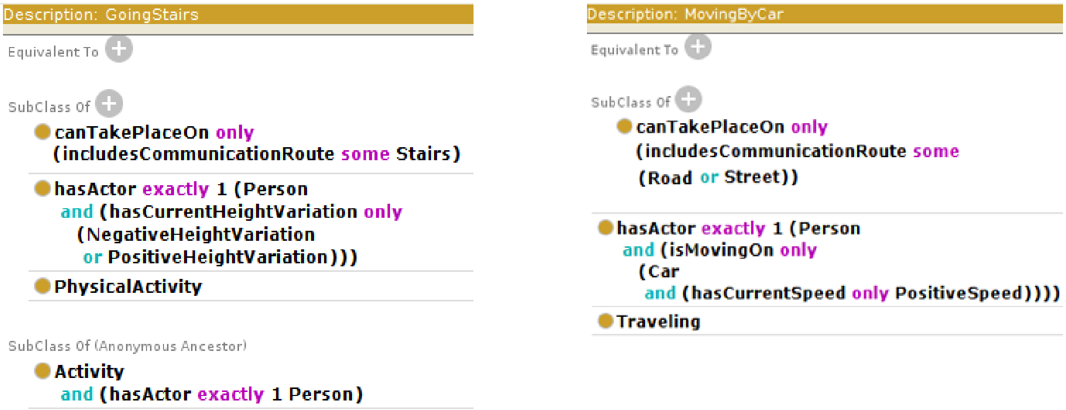
(b) An excerpt of symbolic locations hierarchy

Figure 2: Excerpts of our ontology

Due to the intrinsic open-world assumption of ontological reasoning, we explicitly state the necessary conditions which make activities possible or not possible in a given context. As we will explain later, such constraints are necessary to enable our context-aware refinement which is based on *consistency* reasoning. For instance, the activity `TakingStairs` (Figure 3a) should take place at a location that may have stairs and the person should have a non-negative height variation. Another example is the activity `MovingByCar` (Figure 3b): our ontology enforces that it should take place in an outdoor location which includes a road or a street and that the car's speed should be positive.

[1] `https://protege.stanford.edu/` (Accessed on 2020-02-19)

[2] `https://developers.google.com/places/supported_types` (Accessed on 2020-02-19)



```
Description: GoingStairs
Equivalent To

SubClass Of
  canTakePlaceOn only
   (includesCommunicationRoute some Stairs)
  hasActor exactly 1 (Person
   and (hasCurrentHeightVariation only
     (NegativeHeightVariation
      or PositiveHeightVariation)))
  PhysicalActivity

SubClass Of (Anonymous Ancestor)
  Activity
   and (hasActor exactly 1 Person)
```

(a) Definition of the activity "taking stairs"

```
Description: MovingByCar
Equivalent To

SubClass Of
  canTakePlaceOn only
   (includesCommunicationRoute some
   (Road or Street))

  hasActor exactly 1 (Person
   and (isMovingOn only
    (Car
     and (hasCurrentSpeed only PositiveSpeed))))
  Traveling
```

(b) Definition of the activity "moving by car"

Figure 3: Examples of activity definitions in our ontology

4.2.2. Translating context data into ontological facts

Context data collected by the mobile devices are automatically mapped to the respective ontological concepts by a specifically designed middleware. This middleware encodes the rules that are necessary to transform raw context data into high-level axioms. The majority of context-data can be mapped one-to-one with ontological entities. For example, the user semantic location obtained from a dedicated service is directly mapped by the middleware to the respective ontological fact (see Example 4.2).

Considering scalar values, the middleware discretizes them taking into account the entities covered by the ontology. For instance, each user's speed value is mapped to one of the following ontological concepts: `NullSpeed`, `LowSpeed`, `MediumSpeed`, and `HighSpeed`. The specific rules that map a scalar value to an ontological concept are based on ranges of values designed by the knowledge engineer (e.g., speed values greater than 0 km/h and lower than 8 km/h are mapped to `LowSpeed`).

4.2.3. Context reasoning and refinement

For each activity candidate A in the statistical prediction $h(fv)$, our system uses ontological reasoning to determine whether A is consistent or not with the current context. As a first step, CAVIAR adds to the knowledge base an axiom to represent an instance of `Person` which identifies the subject wearing the mobile devices. As a second step, context data collected by the mobile devices are automatically mapped to ontological concepts as described above and then added as axioms to the knowledge base. As a third and final step, since we want to test the consistency of activity A with respect to the current context, CAVIAR adds an axiom which states that the user is performing A.

Example 4.2. *Bob is using CAVIAR. When the context reasoning task is triggered, `Person(Bob)` is added as a fact. Then, context data gathered by mobile devices is analyzed to expand the set of facts. Suppose that a Web service provides the information that Bob is in a park and that the speed value obtained by the GPS sensor is 10 km/h. Those context information are automatically pre-processed to instantiate the following individuals: `Park(place)`, `MediumSpeed(speed)`. Then, the relationships between Bob and context data are added as facts: `hasCurrentSymbolicLocation(Bob, place)`, `hasCurrentSpeed(Bob, speed)`. Finally, in order to test whether the activity `Running` is context-consistent, CAVIAR adds the axioms `Running(currentActivity)` and `isPerforming(Bob, currentActivity)`. The consistency of the set of facts with respect to the domain knowledge will determine if the running activity is consistent according to the current Bob's context.*

We define an activity A *context-consistent* when the axioms created with the observed data as described above are consistent concerning the domain knowledge. Note that the consistency check involves reasoning that is automatically performed in the logic used to specify the ontology (the specific language and reasoner used in CAVIAR are reported in Section 5).

Given the current context C and the marginal probabilities obtained by the semi-supervised classifier $h(fv) = \langle p_1, p_2, \ldots, p_m \rangle$, the goal of context refinement is to exclude those activities which are not *context-consistent* according to C. For each activity class ac_i such that $p_i > 0$, we compute its consistency according to context C as explained above. Each activity that is not *context-consistent* is removed from the probability vector. The refined vector is finally normalized to preserve the properties of a probability distribution. The output is a new refined probability vector $\langle r_1, r_2, \ldots, r_c \rangle$ such that each A_i is a *context-consistent* activity according to C, $0 \leq r_i \leq 1$ and $\sum_i^m r_i = 1$. Note that an activity is usually not *context-consistent* when ontology's necessary constraints discussed in Section 4.2.1 are violated.

Example 4.3. *Continuing Example 4.2, suppose that Bob is actually running. According to the* Incremental Activity Recognition *classifier, the current probability distribution is* 45% *cycling,* 40% *running,* 10% *walking and* 5% *standing. Thanks to a dedicated Web service, it is possible to know that Bob is currently in a pedestrian area of the park where bicycles are not allowed. According to the ontology, cycling is not context-consistent since it should not be performed in pedestrian areas. Hence, the resulting context-refined probability distribution is* 73% *running,* 18% *walking and* 9% *standing.*

4.3. Prediction Confidence Evaluation

The Prediction Confidence Evaluation module is in charge of using context-refined predictions to update the activity model with new labeled samples using active learning. Moreover, it also applies an incremental data balancing technique to further improve the recognition of minority activity classes.

4.3.1. Active learning

In order to update and improve the activity model, we apply an active learning strategy asking a feedback from the user about her current activity when there is uncertainty in the context-refined prediction. In particular, we adopt a state-of-the-art non-parametric method called *VAR-UNCERTAINTY* [33]. This method is based on a threshold θ which is dynamically adjusted over time. Initially, this threshold is initialized to $\theta = 1$. Given a context-refined prediction $\langle r_1, r_2, \ldots, r_c \rangle$, we denote with $r^\star = \max_i r_i$ the probability value of the most likely activity $A^\star$. If $r^\star$ is below θ, we consider the system uncertain about the current activity performed by the user. In this case, an active learning process is started by asking the user to provide the ground truth A^f about the current activity. The feedback A^f is used to update the activity model with a new labeled data sample. When $A^f = A^\star$, it means that the most likely activity was actually the one performed by the user, and hence the threshold θ is decreased to reduce the number of questions. On the other hand, when $A^f \neq A^\star$, θ is increased. More details about the VAR-UNCERTAINTY algorithm can be found in [33]. For the sake of usability, when querying the user, CAVIAR only presents a few alternatives taken from the most probable activities.

4.3.2. Incremental data balancing

In our everyday life, some activities are performed with a lower frequency than others (e.g., the amount of time that a subject spends on an elevator is usually less than the time he/she spends walking). Hence, updating the incremental classifier with new labeled data samples without taking into account this aspect may lead to an unbalanced classification model, and subsequently to a poor recognition rate on the "minority" activity classes. For this reason, our ontology also describes (using OWL2 properties) which activity classes are known to represent "minority" activities according to common-sense knowledge.

In order to balance the activity model in an incremental fashion, CAVIAR applies a novel online oversampling strategy based on the SMOTE technique [34]. Given M as the set of "minority" activity classes according to the ontology, for each $A \in M$, CAVIAR stores a fixed-size circular buffer B_A that contains β feature vectors having A as their label. Initially, each buffer B_A is filled with β feature vectors randomly extracted from the initial training set used for the model bootstrap (see Section 4.1.2). When CAVIAR triggers a user query originated by a feature vector fv receiving A^f as an answer with A^f in M (i.e., the user is performing a minority activity), the oversampling strategy creates additional synthetic feature vectors. Each of these vectors is computed by perturbing fv according to the SMOTE algorithm with a feature vector randomly chosen from B_{A^f}. Finally, each synthetic feature vector is used as a new labeled data sample to update the activity model, and fv is added to B_{A^f}. Clearly, each buffer may contain feature vectors obtained from different users.

The number of synthetic feature vectors generated by SMOTE is a fixed constant that CAVIAR uses to reinforce minority activity classes in the activity model. In particular, in our experimental setup, we use 4 as a reinforcement constant, since we consider this value appropriate for balancing activities in the target domain. The size of the circular buffers (β) is another constant used in this process. While it would be desirable to generate synthetic labeled data samples considering as many feature vectors as possible, it would be inefficient in terms of storage. In CAVIAR, the value of this constant is $\beta = 100$.

5. Experimental evaluation

In order to evaluate CAVIAR, we implemented a data collection infrastructure to acquire a labeled dataset consisting of both inertial sensor data and context data. Indeed, to the best of our knowledge, there is no publicly available dataset of labeled activities that incorporates rich contextual information. In this section, we describe our experimental setup, the collected dataset, and the results obtained evaluating CAVIAR on the dataset.

5.1. Experimental setup

In our experimental setup, users carry a smartphone in their pants' front pocket and a smartwatch on the dominant hand's wrist. In our data collection, we adopted a Nexus 5x as smartphone and an LG G Watch R as smartwatch. Dedicated Android applications run on those devices to continuously collect and transmit sensor measurements to a Java server, which stores them into a MongoDB database. Both mobile devices communicate to the server every 3 seconds the readings collected from their accelerometer, gyroscope and magnetometer sensors.

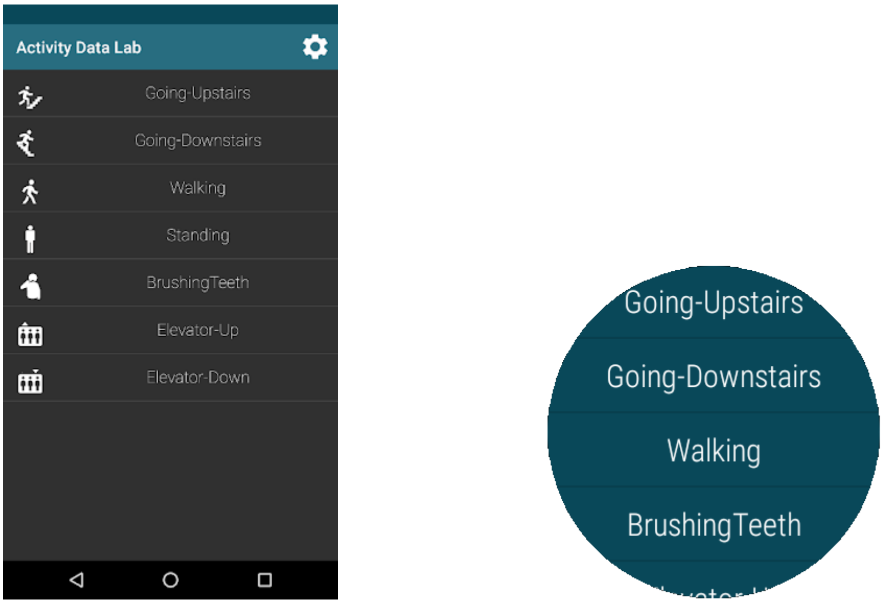
Context data is acquired by the smartphone application considering built-in sensors as well as publicly available web services. The considered built-in sensors are: the barometer measuring height variations, the luminosity sensor, the microphone to obtain the environment's noise level, and the GPS revealing the user's location and speed. The geographical coordinates are sent by the smartphone to the following web services:

- **Google's Places API**: the service returns the most likely semantic places at the location where the user is positioned.

- **OpenWeatherMap** [3]: the service returns the current local weather conditions (e.g., sunny, cloudy, rainy), temperature, wind speed, etc.
- **Bing's Traffic API** [4]: the service returns the nearby traffic situation like road conditions, presence of road works, presence of car accidents, etc.
- **Transitland**[5]: the service returns information about transportation routes and stops close to the user.

The application also collects temporal context like the moment of the day (e.g., morning, afternoon, evening), the day of the week, the season, etc. Every 5 seconds, context data is transmitted to the server.

Besides data acquisition, mobile applications have user-friendly interfaces that allow users to annotate data in real-time. Examples of such interfaces are shown in Figure 4. During the acquisition, we asked the users to use the smartwatch interface to label their activities to speed up the annotation process and, at the same time, to make the acquisitions more realistic.



(a) Smartphone interface

(b) Smartwatch interface

Figure 4: Annotation interfaces

5.2. Dataset description

We acquired a dataset involving 26 volunteers aged between 20 and 28 not involved in this research. They were instructed on how to use the annotation system but not on how to specifically execute the target activities. The activities acquired in this dataset are the following: *walking*, *running*, *standing*, *lying*, *sitting*, *stairs up*, *stairs down*, *elevator up*, *elevator down*, *cycling*, *moving by car*, *sitting on transport*, *standing on transport* and *brushing teeth*.

Those activities have been acquired in different contexts, which include working at the office, going around in the city (Milan), driving, using public transportation, cycling, and staying at home. Overall, we recorded almost 9 hours of labeled and context-rich sensor data ($\sim$ 350 activity instances).

[3] `https://openweathermap.org/` (Accessed on 2020-02-19)
[4] `https://docs.microsoft.com/en-us/bingmaps/rest-services/traffic/` (Accessed on 2020-02-19)
[5] `https://transit.land/` (Accessed on 2020-02-19)

Table 1 summarizes how many minutes of data we acquired for each activity. Note that the dataset is unbalanced, since, as predictable, activities like *taking the elevator* and *brushing teeth* have been executed for a significantly shorter time than others, like *walking*. Nevertheless, this imbalance reflects a realistic duration of activities, thus allowing us to better evaluate the prediction capability of CAVIAR.

Activity	**Minutes**
Standing	52
Sitting	56
Lying	40
Walking	96
Running	24
Cycling	24
Brushing teeth	16
Stairs up	16
Stairs down	16
Elevator up	8
Elevator down	16
Sitting on transport	60
Standing on transport	60
Moving by car	40
Overall	524

Table 1: Number of minutes acquired for each activity

5.3. Results

In the following, we present the results of CAVIAR considering the dataset described above. We use Online Random Forest [35] as classifier, since it is the incremental version of the well-known Random Forest machine learning algorithm, which proved to be one of the most effective classifiers for activity recognition [36]. We take advantage of the Java implementation proposed in [37]. HermiT [38] in combination with the Java OWL API [39] is our OWL2 ontological reasoner.

Since there is no system in the literature to directly compare with, we implemented two variants of CAVIAR. The former is called *Without context*, since it only considers inertial sensor data to recognize activities. In particular, it combines the Incremental Activity Recognition module (see Section 4.1) and the Prediction Confidence Evaluation module (see Section 4.3.1) without applying our context-refinement. Note that *Without Context* can be considered as a baseline, since it is a standard approach for activity recognition [1].

The latter variant of CAVIAR is called *Context as features*. This method, instead of using semantic refinement, incorporates context data directly in the feature vectors generated by the feature extraction mechanism presented in Section 4.1.1. In particular, this method extracts a) statistical features (average, variance, difference between max and min) from *numeric* context data like speed or height variations, and b) binary features for *symbolic* context data (i.e., semantic place, weather condition, proximity to transportation routes, etc.).

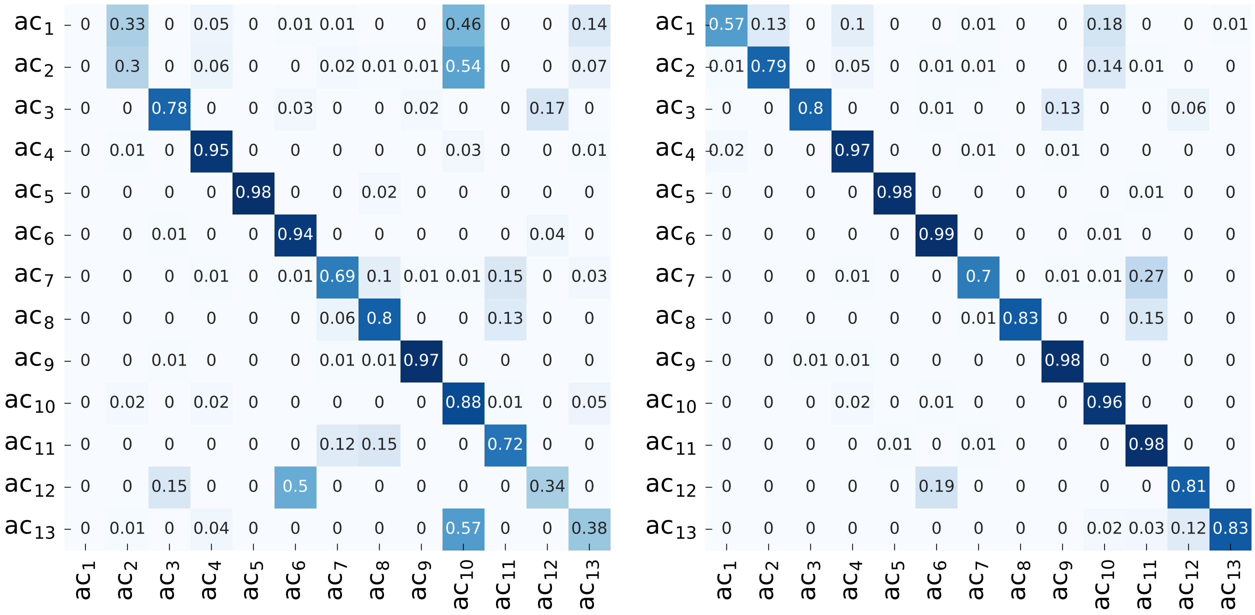
We used a *leave-one-subject-out* cross-validation approach to evaluate and compare CAVIAR with these two variants in terms of recognition rate and the number of subject questions. At each fold, we apply CAVIAR to 25 subjects to collaboratively update the activity model, which is initialized considering 1 minute of labeled data samples for each activity. The data of the remaining subject is used to compute the recognition rate, and the number of questions asked to the subject. In a real application of CAVIAR, users independently perform their activities and concurrently interact with the system. In order to simulate the interleaving of feature vectors coming from different users, the data samples obtained from the 25 subjects are randomly interleaved when given to the classifier.

Table 2 shows the results (in terms of overall F1 score). The results clearly show that context data has a

Activity	Without Context	Context as features	CAVIAR
Elevator up	0.0	0.09	**0.70**
Elevator down	0.31	0.65	**0.83**
Moving by car	0.76	0.80	**0.87**
Brushing teeth	0.77	**0.83**	**0.83**
Running	0.98	0.97	**0.98**
Sitting	0.94	0.96	**0.97**
Going upstairs	0.38	0.45	**0.77**
Going downstairs	0.58	0.81	**0.90**
Cycling	**0.96**	**0.96**	0.93
Standing	0.85	0.95	**0.96**
Walking	0.84	0.89	**0.95**
Sitting transport	0.35	0.62	**0.78**
Standing transport	0.41	**0.97**	0.90
Avg F1	0.62	0.77	**0.88**

Table 2: Recognition rate (F-1 score) of CAVIAR compared with alternative approaches

significant impact on the overall recognition rate. Moreover, context data also allows our method to consider a wider set of activities compared to standard methods which only consider inertial sensors (e.g., *Without Context*). Indeed, activities that are characterized by similar inertial patterns but that are typically executed in very different context conditions can be easily discriminated by CAVIAR. For example, it is evident that activities like going upstairs/downstairs and sitting/standing on transport (which are more difficult to recognize only considering motion patterns) highly benefit from context data. The positive impact of context in reducing confusion between activities is also notable in the confusion matrices reported in Figure 5. From the confusion matrix in Figure 5b we see, for example, that our approach allows the classifier to recognize the *elevator up* activity, while the statistical methods confuse that activity with *standing* since they have very similar motion patterns.



	ac_1	ac_2	ac_3	ac_4	ac_5	ac_6	ac_7	ac_8	ac_9	ac_{10}	ac_{11}	ac_{12}	ac_{13}
ac_1	0	0.33	0	0.05	0	0.01	0.01	0	0	0.46	0	0	0.14
ac_2	0	0.3	0	0.06	0	0	0.02	0.01	0.01	0.54	0	0	0.07
ac_3	0	0	0.78	0	0	0.03	0	0	0.02	0	0	0.17	0
ac_4	0	0.01	0	0.95	0	0	0	0	0	0.03	0	0	0.01
ac_5	0	0	0	0	0.98	0	0	0.02	0	0	0	0	0
ac_6	0	0	0.01	0	0	0.94	0	0	0	0	0	0.04	0
ac_7	0	0	0	0.01	0	0.01	0.69	0.1	0.01	0.01	0.15	0	0.03
ac_8	0	0	0	0	0	0	0.06	0.8	0	0	0.13	0	0
ac_9	0	0	0.01	0	0	0	0.01	0.01	0.97	0	0	0	0
ac_{10}	0	0.02	0	0.02	0	0	0	0	0	0.88	0.01	0	0.05
ac_{11}	0	0	0	0	0	0	0.12	0.15	0	0	0.72	0	0
ac_{12}	0	0	0.15	0	0	0.5	0	0	0	0	0	0.34	0
ac_{13}	0	0.01	0	0.04	0	0	0	0	0	0.57	0	0	0.38

(a) Without context

	ac_1	ac_2	ac_3	ac_4	ac_5	ac_6	ac_7	ac_8	ac_9	ac_{10}	ac_{11}	ac_{12}	ac_{13}
ac_1	0.57	0.13	0	0.1	0	0	0.01	0	0	0.18	0	0	0.01
ac_2	0.01	0.79	0	0.05	0	0.01	0.01	0	0	0.14	0.01	0	0
ac_3	0	0	0.8	0	0	0.01	0	0	0.13	0	0	0.06	0
ac_4	0.02	0	0	0.97	0	0	0.01	0	0.01	0	0	0	0
ac_5	0	0	0	0	0.98	0	0	0	0	0	0.01	0	0
ac_6	0	0	0	0	0	0.99	0	0	0	0.01	0	0	0
ac_7	0	0	0	0.01	0	0	0.7	0	0.01	0.01	0.27	0	0
ac_8	0	0	0	0	0	0	0.01	0.83	0	0	0.15	0	0
ac_9	0	0	0.01	0.01	0	0	0	0	0.98	0	0	0	0
ac_{10}	0	0	0	0.02	0	0.01	0	0	0	0.96	0	0	0
ac_{11}	0	0	0	0	0.01	0	0.01	0	0	0	0.98	0	0
ac_{12}	0	0	0	0	0	0.19	0	0	0	0	0	0.81	0
ac_{13}	0	0	0	0	0	0	0	0	0	0.02	0.03	0.12	0.83

(b) CAVIAR

Figure 5: Comparison of confusion matrices. ac_1 = Elevator Up, ac_2 = Elevator Down, ac_3 = Moving by Car, ac_4 = Brushing Teeth, ac_5 = Running, ac_6 = Sitting, ac_7 = Going upstairs, ac_8 = Going Downstairs, ac_9 = Cycling, ac_{10} = Standing, ac_{11} = Walking, ac_{12} = Sitting Transport, ac_{13} = Standing Transport.

In general, the fact that CAVIAR outperforms the *Context as features* approach shows the value of context reasoning with common knowledge with respect to using raw context data in a statistical approach.

On the negative side, we observe that the recognition rate of CAVIAR on the *cycling* activity is lower than the ones obtained by the other approaches. Indeed, as Figure 5 shows, this activity is often confused by CAVIAR with *moving by car*. This is due to the fact that the available context data that characterizes those activities is similar (e.g., they are both performed outdoor in the city traffic, with variable speed, etc.).

Besides the recognition rate, a crucial evaluation parameter is the number of questions triggered by the system, since it has a significant impact on usability. As Figure 6 shows, CAVIAR generates a significantly lower number of questions (6%) compared to *Without context* (22%) and *Context as features* (16%).

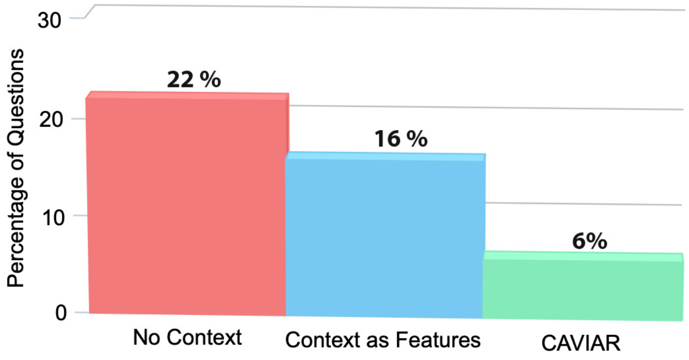


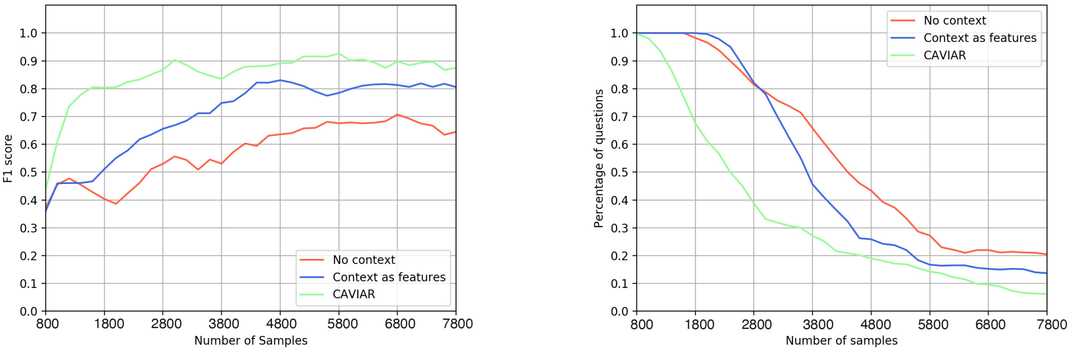
Figure 6: Percentage of triggered queries of CAVIAR compared with alternative approaches

Indeed, our semantic refinement technique exploits the ontology to remove unlikely activities from the prediction, thus significantly increasing the confidence of the remaining activities. Results indicate that CAVIAR should provide a much better user experience by limiting the number of times a user is interrupted with a question.

In order to evaluate how the recognition rate and the number of triggered questions evolve over time, we used the method proposed in [40]. First, we initialize the model as described above, considering 1 minute of samples for each activity. Then, we classify each data sample of the dataset (considering all 26 subjects) using the current recognition model and, depending on the prediction's confidence, we update the model. The classification's output (i.e., the resulting most likely activity), and the corresponding ground truth are collected in sliding windows of 800 samples with an overlap of 75% to periodically compute the overall F-1 score, and the percentage of triggered questions. Samples coming from different users are randomly interleaved. Figure 7 shows the evolution of the F-1 score and the number of questions of CAVIAR compared to the other two considered approaches. Compared to *Without Context* and *Context as features*, CAVIAR quickly reaches high recognition rates, and a significantly lower number of questions.

In order to further show the impact of context reasoning on activity recognition, we also evaluated our system considering different sets of activities. Figure 8 shows how our system performs on a restricted set of simple physical activities that are considered in the majority of related works. We observe that those activities are poorly characterized by the context which surrounds the user, while their movement patterns can be easily discriminated by purely statistical models.

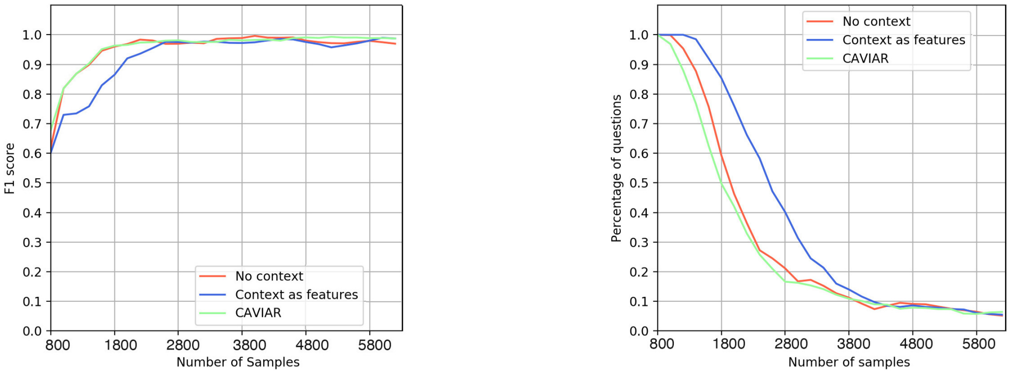
Indeed, the results show that the recognition rate and the number of questions reached by CAVIAR are similar to the ones obtained by the *Without Context* approach. It also strikes out that *Context as features* method has a slower learning improvement, due to the fact that additional context features add complexity to the semi-supervised activity model.



(a) F1 score

(b) percentage of questions

Figure 7: Evolution of the recognition model over time. Considered activities: Running, Sitting, Cycling, Standing, Walking, Elevator up, Elevator down, Going Upstairs, Going Downstairs, Brushing Teeth, Moving by car, Sitting transport, Standing transport

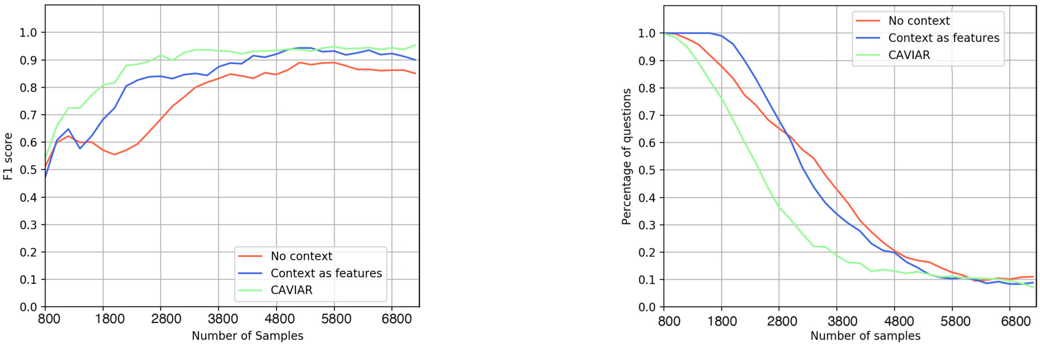


(a) F1 score

(b) percentage of questions

Figure 8: Evolution of the recognition model over time. Considered activities: Running, Sitting, Cycling, Standing, Walking

Finally, we also considered an "intermediate" set of activities. That set includes more context-dependent activities, like *moving by car*, *brushing teeth*, *elevator* and *stairs*. The results are shown in Figure 9.



(a) F1 score

(b) percentage of questions

Figure 9: Evolution of the recognition model over time. Considered activities: Running, Sitting, Cycling, Standing, Walking, Elevator, Stairs, Brushing Teeth, Moving by car

From the plots, it emerges that context-data has a high impact both on the recognition rate and on the number of questions. Indeed, *Without context* struggles to reach the performance of context-based solutions.

Moreover, CAVIAR reaches high recognition rates and a low number of questions very quickly with respect to *Context as features*. This confirms that context-data allows CAVIAR to expand the set of recognizable activities.

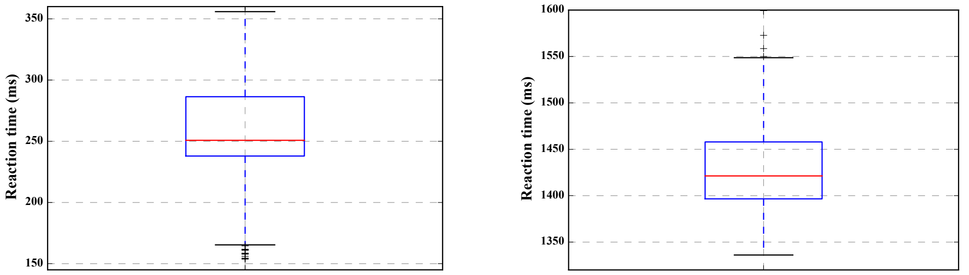
Note that, for all the considered set of activities, CAVIAR maintains the same performance trend while the one of *Context as features* degrades when the complexity of the considered activities increases. Overall, the experimental results suggest that CAVIAR is more scalable with respect to the set of considered activities, and more effective in reducing the number of triggered questions.

5.4. Reaction time

In order to understand if CAVIAR could be practically deployed with current technology for real-time activity recognition, we measured statistics about its *reaction time* defined as the time required to obtain the predicted activity given a segment of sensor readings.

The most likely deployment is cloud-based with a hypothetical CaaS (CAVIAR as a Service) running most of the server-side modules described in our architecture, except for segmentation and feature extraction that may be running on the smartphone. In our experiment, we use a OnePlus 6 smartphone connected to a 4G network and a Linux-based machine with an Intel(R) Core(TM) i7-6700 CPU (3.40GHz) and 16GB of RAM as back-end server. The reaction time is computed considering the computation time (both client- and server-side) and the network communication delay. The boxplot in Figure 10a shows that the overall average reaction time of CAVIAR is only 250ms, with a small variance. We observed that network communication introduces, on average, a delay of 38ms, while the task that required the highest computational effort is ontological reasoning, which took on average 150ms. As for all cloud services, network latency may have an impact on quality of service, but for real-time services like this one, edge computing may offer a solution.

We also investigated the feasibility, in terms of reaction time, of running the whole CAVIAR framework on the mobile device of the user as a local incremental classifier. We performed this experiment by porting all the Java modules on the Android platform, and running a set of time performance tests on a portion of the dataset. The boxplot in Figure 10b shows that CAVIAR takes on average one second and a half to infer the most likely activity from each segment. As expected, the critical task in terms of computing time is ontological reasoning, which took on average 1300ms. Since we considered a segment size of 4 seconds, even this on-device deployment seems to be feasible in terms of reaction time, especially considering the expected evolution of mobile hardware capabilities.



(a) Reaction time of CAVIAR running as a cloud-based service

(b) Reaction time of CAVIAR running locally on a smartphone

Figure 10: CAVIAR reaction time

6. Discussion

6.1. Context-data as features in machine learning versus knowledge-based context reasoning

Our results show the positive impact of context in semi-supervised activity recognition, but they also show that the recognition rate of CAVIAR with respect to the one achieved by a pure machine learning approach using context as features is only 11% higher in terms of overall F1 score. While we believe that this improvement will further increase when considering a larger set of context-dependent activities, the other advantages of our knowledge-based method are: a) the reduced number of queries needed to reach that high recognition rate, and b) a significantly faster learning curve. These aspects improve usability in a real application scenario.

We also believe that our approach is more flexible in terms of the availability of context data. Since context sources may not always be available, using context as features may lead to missing values in the feature vectors used to update the classifier, which in turn may negatively affect the recognition rate. The flexibility of CAVIAR seems also superior when additional types of context become available while the system is deployed. In the case of using context as features, the model requires re-training from scratch with new labeled data. In our method, considering new context data simply means extending the ontology. Moreover, the ontology can also be periodically revised and improved independently from the classifier.

6.2. Knowledge-engineering effort

The knowledge-engineering effort required to design a comprehensive ontology is in general can be very demanding. Indeed, this is a task usually performed by a team of domain experts and knowledge engineers. However, the problem is mitigated by re-using existing ontologies. For instance, in this work we extended the ontology presented in [10].

Moreover, the inevitable incompleteness of manual ontology design, due to the diversity and complexity of context situations and human activities, may be mitigated by exploiting the information acquired through active learning to continuously refine the ontology [41].

6.3. Resource efficiency

While the focus of our contribution is on the potential impact of context reasoning and not on the many engineering aspects that should be addressed in a real deployment, we acknowledge that continuous context acquisition in CAVIAR may have non-negligible costs, especially in terms of energy consumption. Indeed, our system architecture considers mobile devices that generate data streams from several sources (e.g., inertial sensors, GPS, web services).

Resource efficiency is still an open research problem in activity recognition [1]. In addition to the rapid evolution of battery technology that may mitigate this issue, there are research proposals for energy-saving techniques that significantly reduce the amount of data produced by sensors, with a small negative impact on the recognition rate [42, 43].

Even considering deployment models in which the main processing modules are running on the mobiles, there are studies that propose specifically designed lightweight machine learning frameworks [44]. Similarly, lightweight semantic models can be adopted to improve Semantic Refinement [45]. As future work we plan to experiment with the rich variety of resource-aware approaches in the literature to make CAVIAR more practical. In particular, we will investigate the well-known trade-off between resource efficiency and overall accuracy.

7. Conclusion and future work

In this paper, we proposed CAVIAR, a novel real-time activity recognition framework combining context reasoning and semi-supervised learning. CAVIAR relies on ontological reasoning over context data and activities to refine the machine learning predictions, and on active learning to incrementally refine the statistical activity model. Our results, based on a real dataset, show that the refinement of activity predictions due to context reasoning leads to an overall improvement of more than 25% over a baseline that does not take into account context data. We also compared CAVIAR with a pure machine learning approach considering the same context data in the form of additional features. CAVIAR still reaches higher recognition rates and, more importantly, it also triggers a significantly lower number of queries for active learning.

A limitation of our approach is the rigid formalism for knowledge representation and reasoning that cannot take into account the intrinsic uncertainty and incompleteness of common knowledge and sensor technology. Hence, in future work we plan to evaluate alternative probabilistic formalisms. Another interesting extension may be to consider as context the sequence of recently performed activities. This may be achieved by introducing a form of temporal reasoning.

Finally, CAVIAR could be extended by creating a personalized activity model for each user. Indeed, different subjects may have different physical characteristics and habits and personalization may lead to a significant improvement in the recognition of certain activities. This extension not only requires studying how to personalize the machine learning model, but also how to learn user-dependent correlations between context and activities.

Acknowledgment

The authors would like to thank the former students Giovanni Laganà, Matteo Zampatti, Chiara Mariani, Federica Dondi and Alessio Anghileri which contributed to this work with dataset acquisition and software implementation.

References

[1] O. D. Lara, M. A. Labrador, et al., A survey on human activity recognition using wearable sensors., IEEE Communications Surveys and Tutorials 15 (3) (2013) 1192–1209.

[2] O. Köpüklü, A. Gunduz, N. Kose, G. Rigoll, Real-time hand gesture detection and classification using convolutional neural networks, in: 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), IEEE, 2019, pp. 1–8.

[3] J. R. Kwapisz, G. M. Weiss, S. A. Moore, Activity recognition using cell phone accelerometers, ACM SigKDD Explorations Newsletter 12 (2) (2011) 74–82.

[4] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, S. Krishnaswamy, Activity recognition with evolving data streams: A review, ACM Computing Surveys (CSUR) 51 (4) (2018) 71.

[5] H. S. Hossain, M. A. A. H. Khan, N. Roy, Active learning enabled activity recognition, Pervasive and Mobile Computing 38 (2017) 312–330.

[6] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, S. Krishnaswamy, Adaptive mobile activity recognition system with evolving data streams, Neurocomputing 150 (2015) 304–317.

[7] B. Longstaff, S. Reddy, D. Estrin, Improving activity classification for health applications on mobile devices using active and semi-supervised learning, in: Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2010 4th International Conference on Pervasive Computing Technologies for Healthcare, IEEE, 2010, pp. 1–7.

[8] M. Stikic, K. Van Laerhoven, B. Schiele, Exploring semi-supervised and active learning for activity recognition, in: 2008 12th IEEE International Symposium on Wearable Computers, IEEE, 2008, pp. 81–88.

[9] L. Liao, D. Fox, H. Kautz, Location-based activity recognition, in: Advances in Neural Information Processing Systems, 2006, pp. 787–794.

[10] D. Riboni, C. Bettini, COSAR: Hybrid reasoning for context-aware activity recognition, Personal and Ubiquitous Computing 15 (3) (2011) 271–289.

[11] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, P. J. Havinga, A survey of online activity recognition using mobile phones, Sensors 15 (1) (2015) 2059–2085.

[12] N. Györbíró, Á. Fábián, G. Hományi, An activity recognition system for mobile phones, Mobile Networks and Applications 14 (1) (2009) 82–91.

[13] L. Sun, D. Zhang, B. Li, B. Guo, S. Li, Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations, in: International conference on ubiquitous intelligence and computing, Springer, 2010, pp. 548–562.

[14] Y. Mohammad, K. Matsumoto, K. Hoashi, Primitive activity recognition from short sequences of sensory data, Applied Intelligence 48 (10) (2018) 3748–3761.

[15] A. Bulling, U. Blanke, B. Schiele, A tutorial on human activity recognition using body-worn inertial sensors, ACM Computing Surveys 46 (3) (2014) 33:1–33:33.

[16] D. J. Cook, K. D. Feuz, N. C. Krishnan, Transfer learning for activity recognition: A survey, Knowledge and Information Systems 36 (3) (2013) 537–556.

[17] Y. Kwon, K. Kang, C. Bae, Unsupervised learning for human activity recognition using smartphone sensors, Expert Systems with Applications 41 (14) (2014) 6067–6074.

[18] D. Trabelsi, S. Mohammed, F. Chamroukhi, L. Oukhellou, Y. Amirat, An unsupervised approach for automatic activity recognition based on hidden markov model regression, IEEE Transactions on Automation Science and Engineering 10 (3) (2013) 829–835.

[19] M.-S. Lee, J.-G. Lim, K.-R. Park, D.-S. Kwon, Unsupervised clustering for abnormality detection based on the tri-axial accelerometer, ICCAS-SICE 2009 (2009) 134–137.

[20] D. Guan, W. Yuan, Y.-K. Lee, A. Gavrilov, S. Lee, Activity recognition based on semi-supervised learning, in: Embedded and Real-Time Computing Systems and Applications, 2007. RTCSA 2007. 13th IEEE International Conference on, IEEE, 2007, pp. 469–475.

[21] Y.-S. Lee, S.-B. Cho, Activity recognition with android phone using mixture-of-experts co-trained with labeled and unlabeled data, Neurocomputing 126 (2014) 106–115.

[22] E. Hoque, J. Stankovic, Aalo: Activity recognition in smart homes using active learning in the presence of overlapped activities, in: Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2012 6th International Conference on, IEEE, 2012, pp. 139–146.

[23] T. Miu, P. Missier, T. Plötz, Bootstrapping personalised human activity recognition models using online active learning, in: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, IEEE, 2015, pp. 1138–1147.

[24] T. Huynh, B. Schiele, Towards less supervision in activity recognition from wearable sensors, in: 2006 10th IEEE International Symposium on Wearable Computers, Citeseer, 2006, pp. 3–10.

[25] N. D. Rodríguez, M. P. Cuéllar, J. Lilius, M. D. Calvo-Flores, A survey on ontologies for human behavior recognition, ACM Computing Surveys (CSUR) 46 (4) (2014) 43.

[26] U. Akdemir, P. Turaga, R. Chellappa, An ontology based approach for activity recognition from video, in: Proceedings of the 16th ACM international conference on Multimedia, ACM, 2008, pp. 709–712.

[27] Ö. Yürür, C. H. Liu, Z. Sheng, V. C. Leung, W. Moreno, K. K. Leung, Context-awareness for mobile sensing: A survey and future directions, IEEE Communications Surveys & Tutorials 18 (1) (2016) 68–93.

[28] I. Natal, L. Correia, A. C. Garcia, L. Fernandes, Efficient out-of-home activity recognition by complementing gps data with semantic information, First Monday 24 (11) (2019). `doi:10.5210/fm.v24i11.9971`.
URL `https://firstmonday.org/ojs/index.php/fm/article/view/9971`

[29] S. Saguna, A. Zaslavsky, D. Chakraborty, Complex activity recognition using context-driven activity theory and activity signatures, ACM Transactions on Computer-Human Interaction (TOCHI) 20 (6) (2013) 32.

[30] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, D. Riboni, A survey of context modelling and reasoning techniques, Pervasive and Mobile Computing 6 (2) (2010) 161–180.

[31] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, I. Rojas, Window size impact in human activity recognition, Sensors 14 (4) (2014) 6474–6499.

[32] I. Guyon, A. Elisseeff, An introduction to feature extraction, in: Feature extraction, Springer, 2006, pp. 1–25.

[33] I. Žliobaitė, A. Bifet, B. Pfahringer, G. Holmes, Active learning with drifting streaming data, IEEE transactions on neural networks and learning systems 25 (1) (2013) 27–39.

[34] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: synthetic minority over-sampling technique, Journal of artificial intelligence research 16 (2002) 321–357.

[35] A. Saffari, C. Leistner, J. Santner, M. Godec, H. Bischof, On-line random forests, in: Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 1393–1400.

[36] T. Sztyler, H. Stuckenschmidt, 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, 2016, pp. 1–9.

[37] T. Sztyler, H. Stuckenschmidt, Online personalization of cross-subjects based activity recognition models on wearable devices, in: Pervasive Computing and Communications (PerCom), 2017 IEEE International Conference on, IEEE, 2017, pp. 180–189.

[38] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, Z. Wang, Hermit: an owl 2 reasoner, Journal of Automated Reasoning 53 (3) (2014) 245–269.

[39] M. Horridge, S. Bechhofer, The owl api: A java api for owl ontologies, Semantic Web 2 (1) (2011) 11–21.

[40] J. Gama, R. Sebastião, P. P. Rodrigues, On evaluating stream learning algorithms, Machine learning 90 (3) (2013) 317–346.

[41] G. Civitarese, C. Bettini, T. Sztyler, D. Riboni, H. Stuckenschmidt, newnectar: Collaborative active learning for knowledge-based probabilistic activity recognition, Pervasive and Mobile Computing 56 (2019) 88–105.

[42] X. Qi, M. Keally, G. Zhou, Y. Li, Z. Ren, Adasense: Adapting sampling rates for activity recognition in body sensor networks, in: 2013 IEEE 19th Real-Time and Embedded Technology and Applications Symposium (RTAS), IEEE, 2013, pp. 163–172.

[43] J. Liu, B. Priyantha, T. Hart, H. S. Ramos, A. A. Loureiro, Q. Wang, Energy efficient gps sensing with cloud offloading, in: Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, 2012, pp. 85–98.

[44] N. D. Lane, S. Bhattacharya, P. Georgiev, C. Forlivesi, L. Jiao, L. Qendro, F. Kawsar, Deepx: A software accelerator for low-power deep learning inference on mobile devices, in: 2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), IEEE, 2016, pp. 1–12.

[45] M. Bermudez-Edo, T. Elsaleh, P. Barnaghi, K. Taylor, Iot-lite: a lightweight semantic model for the internet of things and its use with dynamic semantics, Personal and Ubiquitous Computing 21 (3) (2017) 475–487.