Pervasive and Mobile Computing 51 (2018) 150–159

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Cross-environment activity recognition using a shared semantic vocabulary

Zachary E. Wemlinger *, Lawrence B. Holder

School of Electrical Engineering & Computer Science, Washington State University, P.O. Box 642752, Pullman, WA 99164-2752, United States

ARTICLE INFO

Article history:
Received 3 May 2018
Received in revised form 2 October 2018
Accepted 10 October 2018
Available online xxxx

Keywords:
Smart environments
Semantics
Activity recognition

ABSTRACT

Effectively recognizing activities in smart environments requires either matching sensors to semantic models or labeled training data from the target environment for machine learning. Combining knowledge-driven and data-driven approaches improves activity recognition (AR) by providing the benefits of each while also mitigating their challenges. In this paper we present the Semantic Cross-Environment Activity Recognition (SCEAR) system which is a novel method for creating semantic feature spaces and enables data-driven AR systems to transfer AR models across environments. We evaluate SCEAR using 22 datasets from real-world smart environments. Transferred model performance is compared to models trained in the target environment and shown to provide a 39% average per-class improvement.

1. Introduction

The current state-of-the-art in data-driven Activity Recognition (AR) in smart environments utilizes feature spaces which are specific to a single smart environment. As a result, to recognize activities in a new environment, one must utilize transfer learning or data must be captured and annotated within that environment before activities can be recognized. Gathering and annotating data increases the cost of deploying smart environments and also adds a delay between deployment and effective assistance. Transfer learning for AR is still an active area of research which can take advantage of our approach for increased effectiveness.

Our contribution in this work is the Semantic Cross-Environment Activity Recognition (SCEAR) system which enables data-driven AR between environments. This is accomplished by projecting raw sensor activity into a semantically-grounded feature space which is common to all environments. Using this feature space we are able to train off-the-shelf machine learning algorithms on annotated data in a source environment and recognize activities in target environments for which we do not have annotated data. SCEAR is a modular system composed of two primary components, an ontology and a set of sensor reasoners.

The ontology, described in detail in Section 3, is used to define the common terminology throughout SCEAR including types of sensors, areas, and objects. Sensor types are used in SCEAR to route sensor events to appropriate sensor reasoners. Types of areas and objects are used as annotations in environment layouts to help connect specific sensors to kinds of things in specific environments. Finally, the types of objects and areas define the output feature spaces used for machine learning.

The second primary component of SCEAR is a set of sensor reasoning modules which compute weights for semantic concepts given sensor observations. Each reasoner accepts as input a filtered subset of events and produces as output a

* Corresponding author.
E-mail address: zewemli@wsu.edu (Z.E. Wemlinger).

https://doi.org/10.1016/j.pmcj.2018.10.004
1574-1192/

real-valued weight which is tied to a specific semantic concept. The output of the set of reasoners defines the stable feature space which is shared between environments. These reasoners are further detailed in Section 5.

Using our approach we train a single binary classifier for each activity and pair of environments. Exploring the results yields new insights into the challenges of modeling resident activity. We can successfully recognize some activities across different sensor platforms and environment layouts. However, we also observe many asymmetries where an activity transfers well from environment A to environment B but does not transfer well from B to A. While SCEAR does not remove the need for having some annotated activity data, it does multiply the impact of such data by allowing it to be applied across different environments.

The remainder of this work is organized as follows. In Section 2 we put our approach in perspective relative to other research into smart environment AR. Section 3 discusses the ontology we have made use of and presents our approach to defining a smart environment using semantic concepts. Next, Section 4 describes how we use a visual approach to environment definition and alignment with our ontology. In Section 5 we present our domain-specific reasoning modules which are used to enable sensor reasoning between heterogeneous sensor platforms. Details regarding dataset usage and label alignment are presented in Section 6. We discuss our experimental methodology in Section 7 and our results in Section 8. We conclude and discuss future directions in Section 9.

2. Related work

There is extensive work in smart environments, with the modern concept of an environment in which there are many simple sensors embedded into the environment presented in [1,2]. Within smart environments, there are two primary branches of research, one which is focused on using data-driven approaches for activity recognition. Examples of data-driven approaches are presented in [3–5]. These contributions presented in these works focus on three areas: sensors, features, and learning algorithms.

With the exception of [3], work in data-driven AR generally does not focus on cross-environment learning. In [3], Cook presents a manual mapping approach in which sensors are associated with rooms in an environment. This showed the potential for cross-environment AR, however, did not provide methods for aligning heterogeneous environments and sensor ecosystems, which is a core contribution of our work here.

The other main branch of AR research is the knowledge-driven approach. Presented in [6], knowledge-driven approaches use well-defined activity matching algorithms combined with highly constrained sensor layouts to recognize activities from predefined templates. These approaches are challenging to apply in unconstrained environments or to activities which are difficult to define a priori. Our approach is primarily data-driven, but makes use of a domain ontology to transform environment-specific sensor activity into a generic feature space for use by machine learning algorithms. As such, our approach can be seen as a hybrid approach between data and knowledge-driven techniques.

Other hybrid approaches to AR are presented in [7–10]. Each of these approaches makes use of ontological representations of activities and sensors combined with probabilistic reasoning approaches to recognize activities. Our approach is distinct from these works in that we make use of ontologies only for the purpose of transforming sensor activity into feature spaces for generic machine learning models. Also, other hybrid approaches have provided experimental results based on datasets which make use a multiple independent volunteers performing the same activities in the same environment. Our evaluation makes use of data collected from different individuals in different environments, which allows us to better estimate performance in real-world scenarios. We expect that future research will find opportunities to improve our SCEAR using techniques from other hybrid approaches as well as improve other approaches by making use of SCEAR for creating feature spaces.

SCEAR makes use of an extended version of the COSE smart home ontology originally presented in [11]. There are several other ontologies for smart homes available including: DogOnt [12] and DomoML [13], ActiveO [7], the ontology discussed in [14], and CONSERT [15]. During the course of research we found that the concepts we have made use of are generally present in most smart home ontologies, though existing ontologies were missing a few concepts. For ontologies which utilize semantic web technologies, such as COSE, DogOnt, and ActivO, sharing and extending the ontology is a trivial task. As such we chose to extend the COSE ontology for our work here. However, we do not see the ontology as a novel contribution and our approach is compatible with a wide range of existing ontologies. Also, SCEAR does not rely on the reasoning capabilities of OWL and instead implements domain-specific reasoners which key off the taxonomic structure in the ontology.

The USMART system described in [16] also makes extensive use of ontological model for activity recognition. While the authors do not specifically discuss using their methods across environments, the modeling choices should allow extension into cross-environment AR.

In [17] Rashidi presents an approach for cross-environment activity recognition which uses a novel activity discovery method to learn activity templates and then computes sensors alignments using a semi-EM algorithm. For each unique combination of source and target environment, the approach in [17] calculates a complete pairwise mapping between sensors. SCEAR differs by explicitly transforming sensor activity to a common semantically-grounded feature space which removes the need to map individual sensors in one environment to sensors in a different environment. This enables us to address sensor fusion and enables cross-environment AR to be applied across significantly different sensor platforms.

Similarly, transfer learning has also been applied to the problem of cross-environment AR. In [18], Feuz and Cook describe a Feature Space Remapping (FSR) technique which can be used to map sensors in one environment to sensors in another environment. Unlike [17], this approach uses features of sensors to calculate alignment between source and target environments, and assumes the semantic meaning of features is not known. SCEAR clearly differs by explicitly defining the semantics of the feature space, which allows SCEAR to be more interpretable and avoids fundamental challenges of mapping semantically incompatible domains.

Also, the work presented in [19] demonstrates methods for clustering a labeled activity into different homogeneous subgroups. This may be relevant to future work aimed at understanding classification symmetries and transferability between environments.

We anticipate that using the features from our approach, transfer learning can be simplified with an emphasis on mapping between individuals and environments.

3. Smart home ontology

Observable data in smart environments is generated by some set of ambient sensors. Those sensors are placed around the environment with the intent of capturing actions relevant to the activities which will be annotated. Given that the observable space is defined by sensors specific to an environment, we need to map these observables into a meaningful shared semantic space. Specifically, we attempt to represent the original physical space in which the resident activity was observed and then describe that space using common ontological terms. We have extended and refined the CASAS Ontology for Smart Environments (COSE) [11] with concepts needed to represent the environments used by our approach.

The ontology is organized such that some concepts have multiple parents, thus forming a lattice rather than a taxonomic tree. We have made extensive use of unary relationships, inheritance, and have found relatively few applications for binary relations such as data and object attributes. In order to extend existing ontological reasoning systems for AR, we have designed a framework for dynamically applying reasoning modules for use in sensor reasoning, which is discussed further in Section 5.

Our ontology is separated into four primary concepts: *Functional Space*, *Household Item*, *Sensor*, and *Sensor Attributes*. Functional spaces use common terminology such as *Kitchen*, *Living room*, *Bedroom*, and *Bathroom*. Household items are comprised of the fixed objects around an environment, such as furniture, plumbing, doors, and appliances. Our approach can be extended to make use of such objects by adding these to the ontology and implementing the necessary sensor reasoning modules. Objects are also categorized based on the type of space in which they are expected to be located, for example, the class *Toilet* is a subclass of the concept *Bathroom Objects*.

The *Sensor* and *Sensor Attributes* portions of the ontology are used by domain reasoners, along with a visual alignment ontology, to transform sensor activity into concepts represented by some subtype of the *Functional Space* or *Household Item* types.

4. Visual ontology

Defining an environment in ontological terms can be a laborious and counter-intuitive task which requires training and expertise to accomplish. Given our desire to limit the burden on human annotators and experts when it comes to deploying AR systems, we have developed a visual representation approach which significantly reduces the effort required to create a structured definition of a smart environment.

Our visual representation of an environment is based on a template image created using Scalable Vector Graphics (SVG), which is a popular vector graphics format. We create an SVG layout of each physical environment using a predefined set of template objects which are aligned to the underlying ontology. Laying out each environment is a short manual process which takes approximately 30 min per environment. The SVG is parsed to create a structured representation of each environment and ray-tracing is used to determine which objects are within the line-of-sight of each sensor (see Fig. 1).

One important caveat for our work here is that we do not have physical access to the environments used for evaluation and have not been able to fully verify the accuracy of our environment models. It is possible that there are some errors in our layouts which have partially limited our results. Also, our layouts do no have precise physical measurements of an environment. As the remainder of this paper demonstrates, these approximate models can be used effectively for activity recognition, however more precise models may be more helpful.

5. Sensor reasoning

To apply our ontology in a smart environment, we must create a sensor reasoning system which takes as input raw sensor events and outputs features aligned with the semantic concepts in the ontology. This output will then be used by machine learning models to recognize activities.

Unlike established knowledge-driven approaches to AR, we are not performing symbolic reasoning. The reasoning here is a set of domain specific modules designed to reason over sensor events in the context of an environment and output estimates regarding the likelihood that a resident is interacting with a area or object.

The input to each reasoning module is the state of all sensors in the environment. Each module selects a subset of these sensors to reason over. The output of each module is a set of tuples which describe the semantic state of the environment. Each tuple is composed of 4 elements ($object, type, phenomenon, value$). The *object* is a unique identifier for a specific object

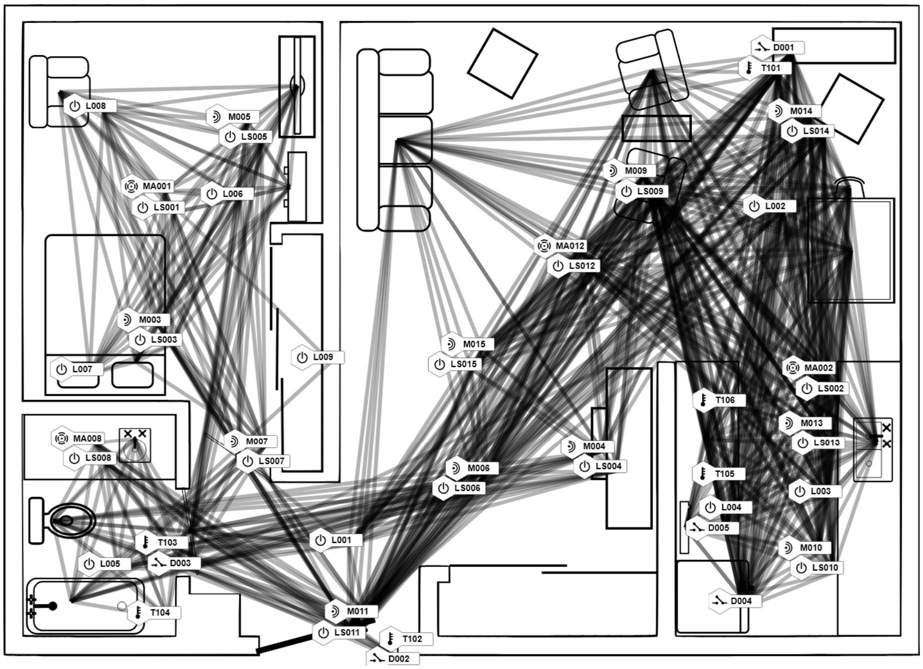


Fig. 1. Example of item connectivity graph using the HH115 environment.

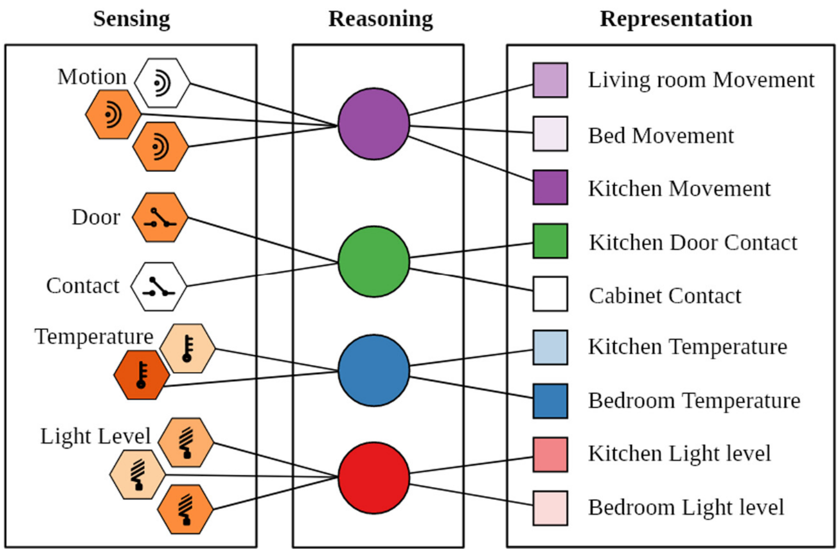
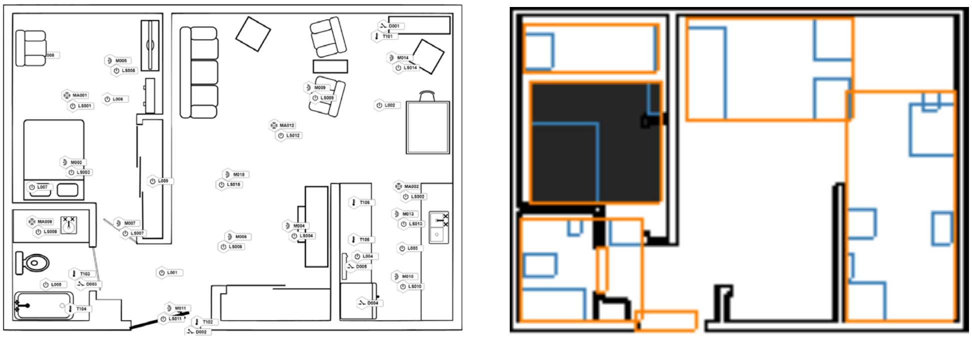


Fig. 2. Illustration of information flow.

in a specific environment, this is ignored when reasoning across multiple environments. The *type* is the semantic type of the *object* and is a concept in the ontology. The *phenomenon* is a semantic concept which is used to reflect the physical phenomenon which was detected by the underlying sensor. There are currently four phenomena we model: Movement, Contact, Temperature, and Light. The *value* is a scalar value which reflects an estimate that the semantic concept is relevant to the resident's current activity.

Fig. 2 provides a simple illustration of this system. The left side of the figure represents sensor events, which are routed to the appropriate reasoning module. Each reasoning module controls a portion of the output feature space and fills in the values there based on the current sensor states and environment structure.

The following sections describe the current set of reasoning modules.



(a) Layout for HH115 (b) Sleeping in HH115

Fig. 3. Example activity representations of the HH115 environment. The environment layout with sensors is shown on the left. On the right we visualize a moment out of a *Sleeping* activity. Walls are shown in black, objects in blue, and areas in orange. The darkened area indicates activity in the bedroom.

5.1. Motion detectors

Passive Infrared (PIR) motion detectors make up the bulk of the sensors available in the datasets provided by the CASAS group at WSU. PIR sensors are activated by movement and remain active for a short period before turning off; continued movement reactivates the sensor. PIR sensors used in smart environments are calibrated to detect humans. Pets and small animals can be detected but with low-reliability.

Now, let η be a location in the environment and s be some PIR sensor. We wish to estimate the chance that a resident is at location η given an observation of sensor s. First we use ray-tracing to determine the areas of the environment which are visible to s. Ray-tracing produces a masking function $M(\eta,\ s)$ as shown in Eq. (1).

$$M(\eta,\ s) \Rightarrow \begin{cases} 1 \mid \eta \ \textit{visible to } s \\ 0 \mid \eta \ \textit{occluded from } s \end{cases} \tag{1}$$

In addition to ray-tracing, we also model the sensitivity of the sensor s using a scaling constant λ_s, which is based on the type of PIR sensor used. The euclidean distance between s and η, $\Delta(s, \eta)$, is also taken into account. The resulting estimation function is shown in Eq. (2).

$$\textit{estimate}(\eta \mid s) = M(\eta,\ s)\lambda_s \Delta(s, \eta) \tag{2}$$

At any point in time, there can be multiple sensors active in an environment. We denote the set of all active PIR sensors as $\mathbb{S}$. We fuse these sensors by taking the mean of their estimates, shown in Eq. (3).

$$\textit{Estimate}(\eta \mid \mathbb{S}) = \mathbb{E}[\textit{estimate}(\eta \mid s) \, \forall \, s \ \in \mathbb{S}] \tag{3}$$

The next step is to associate the estimates with physical objects in the environment. Given the environment layout, a bounding box for each object and space in an environment can easily be obtained. Let $\mathbb{B}_o$ be the set of all points in the environment within the bounding box for an item o. We use the mean estimate for $\eta \ \in \ \mathbb{B}_o$ as the estimate for object o.

Fig. 3 shows the environment layout of the HH115 environment and pairs that with a visual representation of the object interaction estimates within the environment while the resident is sleeping. Note in this case the bed is the only object darkened because the only sensor events are from sensors in the bedroom.

5.2. Object sensors

Object sensors are sensors attached to various objects around the environment. These the role of being an object sensor can be played by many types of sensors such as pressure mats and contact sensors. Object sensors are seen as the sole source of evidence with respect to a specific object. For example, when one would be unable to open a door without triggering the door sensor. Our reasoning system allows for individual modules to flag objects which have a single source of information.

When reasoning over object sensors, we take the position of the sensor then find the nearest object of the correct type. For example, door sensors are only assigned to doors or objects which have doors. When the module observes the object sensor, it infers interaction with the assigned object as long as the object sensor is active.

5.3. Environment states

Environment state sensors are the sensors which take periodic measurements of physical properties of an environment. There are two kinds of these sensors present in the datasets we evaluated, the temperature sensor and the light level sensor. Measurements by these sensors may not be directly attributable to resident actions. Measurements from these sensors are associated with the area of the home in which the sensor is located, but is not attributed to a specific object.

Table 1
Activity labels used.

Bathing	Bed_To_Toilet	Breakfast	Changing_Clothes
Cooking	Dinner	Eating	Enter_Home
Exercise	Having_Guest	Housekeeping	Leave_Home
Lunch	Napping	Other	Out_Of_Home
Personal_Hygiene	Read	Relax	Sleeping
Study	Take_Medicine	Talking_On_The_Phone	Toileting
Wake	Wandering	Wash_Dishes	Watch_TV
Work			

6. Cross-environment activity recognition

Once activity in environments is represented using a common vocabulary as described in Section 5, there are several challenges that need to be addressed before cross-environment AR is possible.

First, different datasets have been annotated for different target activities. We have relied on common-sense understandings of the labels to align the 116 different labels across all of the datasets down to 29 common labels. This has included removing some labels which are not used in multiple datasets by mapping those labels to the *other* label, which is a label applied to any activity outside of the set of activities of interest.

Next, we have trained individual models for each unique triple of (*source*, *target*, *activity*). Only feature present in both source and target environments are used to train the model. For environments with multiple residents, the *activity* is labeled if any resident is engaged in that activity.

7. Experimental methodology

Our experiments were designed to establish (a) if activities can be learned across environments and (b) if our semantic representation of activities allows for learning across sensor platforms.

To understand cross-environment Activity Recognition (AR) performance, we have made use of 22 datasets from 20 unique smart home environments. The datasets we have used are: ARAS A and B [20], Aruba [21], Cairo [21], HH102 [21], HH104 [21], HH107 [21], HH110 [21], HH112 [21], HH113 [21], HH115 [21], HH116 [21], HH117 [21], HH118 [21], HH120 [21], HH122 [21], HH123 [21], Kyoto 2010 and Summer 2009 [22], Milan [22], Tulum 2010 and 2009 [23]. All of the datasets used here are publicly available from the research groups which collected the data, see citations for each dataset for details.

For each environment, we create a dataset consisting of the first 30 days of activity in the environment. We train a separate binary classifier for each unique (*source*, *target*, *activity*) triple, which ensures models are never tested in an environment in which they were trained.

7.1. Activity labels

Over the 22 datasets used in this work, there are 116 distinct labels used for activities. We have remapped these down to 29 activity labels for our experiments. Some of the 116 labels are nearly identical except for the part-of-speech chosen for a label, e.g., "bathe" vs. "bathing". While space constraints prevent the enumeration of each mapping, we describe the principles used for mapping below. We do not necessarily suggest that the mappings used are truly optimal, and we did not have direct access to the annotation rubrics used when creating the dataset. However, in general, we found that ambiguities were often resolved easily. The final list of 29 activity labels used is shown in Table 1.

7.2. Machine learning models

Various machine learning models have been used for AR. For this work we have made use of the following algorithms: Naïve Bayes (NB) [1,23], Decision Tree (DT) [4,24], Random Forest (RF) [24–26], and Support Vector Machine (SVM) [4]. While the NB models require no additional parameters, other algorithms have many hyperparameters which can be tuned. For all algorithms, we have used the implementations provided by the Scikit-Learn Python library [27]. The following parameters were chosen based on an initial set of experiments in which we performed a grid search over a wide range of parameters. The selected parameters work the best on average, however we will note that the best performing parameters varied by dataset.

To avoid overfitting, the DT models were trained with the minimum number of samples on a leaf set to 10. The default *gini* score was also used as the split criterion.

The RF models were trained using the following parameters: minimum number of examples on a leaf node *10*, split criterion *gini*, number of trees *32*, and the number of features evaluated at each split is set to the square root of the number of features.

For the SVM models, training time over an entire dataset was too time-intensive to scale to the hundreds of models that needed to be created. Thus, we randomly sampled 10,000 examples from each dataset for training. All other parameters were left at their default values and the kernel used was the Radial Basis Function (RBF) kernel.

7.3. Evaluation

The performance metric used to evaluate AR most often is accuracy, though F1-score is also popular. The class-imbalance in AR datasets can be significant, so we focus on F1-score which is more robust to class-imbalance issues.

To understand performance across many operating points, we could turn to the area under the ROC curve or the Precision–Recall (PR) curve. In [28], Flach describes Precision–Recall-Gain curves and the related area-under-the-curve metric (AUPRG) which provides an alternative to the Precision–Recall curves that is more interpretable and performs well for class-imbalanced problems. AUPRG has the desirable property that it is normalized such that an AUPRG of 0 is assigned to a trivial classifier. Positive AUPRG values a more optimal than a trivial classifier while negative values are less optimal. Exactly what level of performance is needed for AR is not generally agreed upon.

For comparisons with locally trained classifiers, we use a k-fold cross-validation technique in which the first 30 days of a dataset is first separated off into individual days and then sequentially assign days to folds in a round-robin manner. The Machine Learning (ML) algorithms used for cross-environment modeling are those referenced in Section 7.2. The feature set used to train these classifiers is described in [4]. These features represent the state of all sensors in the environment, the time since a sensor was observed, and which sensor was most frequently observed for a sliding window across the dataset.

8. Results

The following sections separate the analysis of our results into two sections. In order to ensure the reader can compare this work with existing approaches, Section 8.1 presents our results as compared to the performance of well established data-driven AR algorithms. In Section 8.2 we more fully explore performance of activity recognition across environments. This area is not well represented in existing literature and is intended to establish a baseline for future work.

Effort was made to align and compare our work with existing semantic approaches to AR. Ultimately, we found that semantically modeling the activities in the datasets used here was infeasible. Successful applications of semantic activity modeling tightly couple the placement of sensors with the underlying ontology and activities which are to be recognized. For example in [6] the authors note that sensors were attached to each object used in an activity which was then modeled using their ontology. The datasets used here were designed with a loose coupling approach where the intent was that activities would be recognized using statistical patterns over sensor outputs.

We also note that, when compared with other methods for ontological modeling of smart environments, our ontology is not intended to provide a novel method for describing a smart environment. Our approach distinguishes itself is by systematically projecting sensor information into a useful semantic space despite the fact that the sensors were not originally placed with the ontology in mind. The fact that we can then apply off-the-shelf machine learning models to this semantic space and recognize activities across a wide range of environments shows expanded opportunities for AR when compared with existing literature.

8.1. Comparisons with local classifiers

When comparing our results with existing approaches it is important to note that the “other” activity was not included in this comparison. The reason is that “other” has no coherent structure or meaning and thus there is no expectation that “other” would transfer from one environment to another which means no cross-environment models were trained to recognize “other”. This significantly reduces the overall F1 performance of the local classifiers because “other” is usually one of best performing classes.

Tables 2 and 3 show the absolute performance of SCEAR as compared to the approach presented in [4]. The performance reported for SCEAR is a weighted average of the best classifiers for each activity from any source environment when applied to the target dataset.

Overall, the results show that SCEAR provides either a 17% or 39% average improvement over locally trained classifiers when using either micro- or macro-averaging. F1-score is only defined for binary classes so for multi-class settings we must average the performance of individual classes. Micro-averaging gives equal weight to each time-step while macro-averaging gives equal weight to each class. In the literature the most frequently used averaging technique is micro-averaging, which is more forgiving when there are many small classes which are difficult to recognize.

We observe two environments, Cairo and Tulum.2009, where SCEAR performs worse than locally trained models. For Cairo, it is clear that neither SCEAR or locally trained models are very effective, which is largely due to annotation challenges associated with pet activity within the home. Tulum.2009 is a multi-resident college apartment where the overall schedule is heavily influence by the semester schedule. This has caused local schedule information to play a significant role in AR.

Often, there are not enough examples of an activity in an environment to learn a robust model for that activity. By making use of information from other environments, SCEAR allows AR systems to leverage a larger pool of labeled activities.

Areas where locally trained classifiers tend to perform well are for activities such as “bed_to_toilet” and “sleeping”. These activities tend to be well represented in the training data. The additional detail provided by local information, e.g., what time a resident normally goes to bed, proves critical to distinguishing these activities from other activities.

Table 2
Absolute performance of SCEAR for the HH datasets. Δ represents the relative increase gained by using SCEAR.

Dataset	Micro-averaged F1			Macro-averaged F1		
	SCEAR	Local	Δ	SCEAR	Local	Δ
hh102	0.34	0.30	13%	0.14	0.13	3%
hh104	0.39	0.31	27%	0.15	0.11	34%
hh107	0.27	0.18	46%	0.13	0.10	30%
hh110	0.44	0.38	14%	0.20	0.14	39%
hh112	0.34	0.28	22%	0.10	0.06	55%
hh113	0.37	0.32	15%	0.16	0.16	1%
hh115	0.30	0.29	6%	0.11	0.07	53%
hh116	0.29	0.27	5%	0.17	0.11	49%
hh117	0.37	0.25	50%	0.16	0.11	48%
hh118	0.31	0.30	4%	0.13	0.09	48%
hh120	0.31	0.28	10%	0.14	0.13	10%
hh122	0.36	0.30	16%	0.11	0.08	39%
hh123	0.32	0.24	36%	0.13	0.07	79%

Table 3
Absolute performance of SCEAR compared to locally trained binary classifiers. Δ represents the relative increase gained by using SCEAR.

Dataset	Micro-averaged F1			Macro-averaged F1		
	SCEAR	Local	Δ	SCEAR	Local	Δ
arasa	0.37	0.34	10%	0.26	0.20	27%
arasb	0.38	0.29	30%	0.24	0.15	56%
aruba	0.55	0.41	33%	0.20	0.12	69%
cairo	0.07	0.09	−26%	0.04	0.06	−32%
kyoto.2010	0.63	0.61	4%	0.10	0.09	19%
kyoto.summer2009	0.51	0.44	16%	0.20	0.19	8%
milan	0.19	0.15	30%	0.13	0.05	150%
tulum.2009	0.45	0.84	−45%	0.29	0.49	−41%
tulum.2010	0.44	0.28	54%	0.19	0.09	107%

Table 4
Percentage of tests with successful activity transfer.

Classifier	Success	Symmetric	Asymmetric
DT	0.78	0.46	0.32
NB	0.74	0.41	0.34
RF	0.77	0.52	0.25
SVM	0.63	0.30	0.33

8.2. Evaluating performance across environments

In total there were 20,664 tests run across all datasets. In the following discussion, we consider a ''successful'' activity transfer to be a transfer with a positive AUPRG score, indicating that it beat the naïve baseline of a constant positive classifier. Note that using AUPRG means we are optimizing for the model which produces the best F1 score. Since the only value in the numerator of the F1 score is the number of true positives, the usual naïve baseline of a constant negative classifier will yield an F1 score of 0. During our evaluations were found frequency asymmetries such that an activity classifier trained in environment A would work well in environment B, however, the reverse was not true. Thus, when discussing activity transfer, we refer to the transfer as either ''Symmetric'' or ''Asymmetric''. Given some activity l and a pair of environments A and B, symmetric transfer is when models trained to recognize l in A are successful in B, and a different model trained in B is successful in A. Asymmetric transfer is when one and only one of the two models are successful. Table 4 summarizes the overall success rate of activity transfer by the classifier as well as showing the symmetries.

As we see in Fig. 4 transfer symmetry varies greatly between activities. The 'out_of_home' activity denotes when a resident is out of the home and is only used in the ARAS environments [20] which share similar sensor layouts. The result is this activity transfers very well, but only between those two environments. Very common activities which transfer well include: *toileting*, *sleeping*, *bed_to_toilet*, *cooking*, and *watch_tv*.

If we consider the transferability between pairs of environments, as shown in Fig. 5, we can see that while there are a couple of datasets, e.g., Cairo and Tulum.2009 which do not transfer well, much of the space is fairly uniform. The HH datasets form a group in the middle of the plot. Given that these environments share very similar sensor layouts, this is not surprising.

We consider the question of which classifier tends to perform best when transferring between environments. The optimal algorithms, in order of percent of optimal pairs, are: RF (67%), DT (21%), SVM (7%), and NB (5%). While RF models generally perform best, it is important to test with different algorithms to find the best fit.

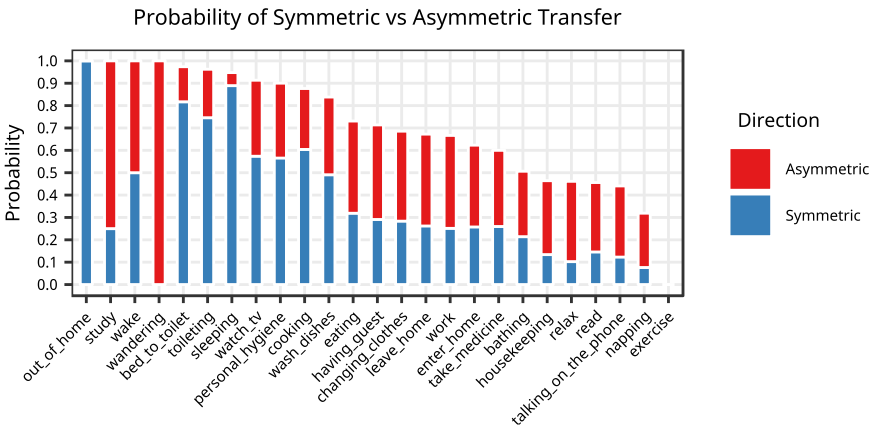


Fig. 4. Probability of successful transfer across all classifiers.

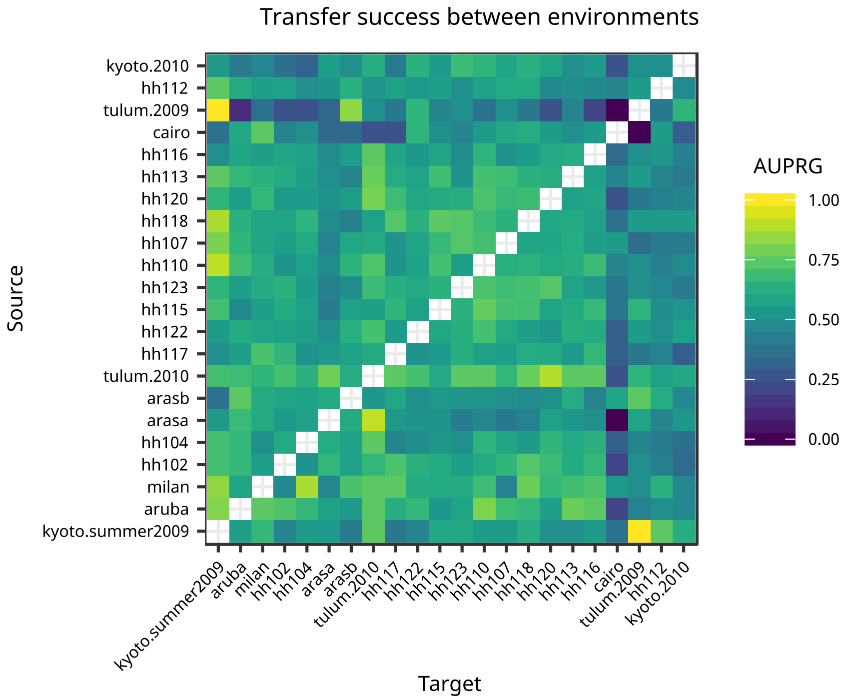


Fig. 5. Transfer between pairs of environments.

9. Conclusions

In this work, we have presented the Semantic Cross-Environment Activity Recognition (SCEAR) system which is a novel approach to perform cross-environment Activity Recognition (AR). Our approach uses an ontology for concept alignment and domain-specific sensor reasoners to infer human actions from sensor activity. When combined with a standard data-driven approach to AR and evaluated on 20,664 AR scenarios where a single activity is transferred between distinct source and target datasets, SCEAR allowed standard Machine Learning (ML) models to out-perform naive baselines in 78% of cases and can provide a 39% relative improvement when compared with locally trained classifiers. We also showed that transferred models can positively predict activities which would be ignored by models trained on local data from the target environment.

During the course of research, we were made acutely aware of the need for better rubrics for activity annotations. The common practice of "common sense" annotating leads to many different, largely overlapping, labels. More standardized labeling practices would be very helpful in driving annotations for data-driven AR and better understanding when each approach is most appropriate.

There are many directions for future research. One area of particular interest is in finding an unsupervised method by which environments can be paired before training supervised models. In this work we have used AUPRG which evaluates model performance across all operating points, finding the optimal operating point requires acquiring labeled examples in the target dataset. This current work does not address how to minimize the number of labels required in the target environment; we look forward to applying techniques from other works to address this issue. We have promising initial results for applying semantic features for training locally within an environment and future work there is focused on how to best use semantics with minimal loss of environment specific details.

References

[1] E.M. Tapia, S.S. Intille, K. Larson, Activity Recognition in the Home Setting Using Simple and Ubiquitous Sensors, Springer Berlin Heidelberg, 2004.

[2] D.J. Cook, S.K. Das, K. Gopalratnam, A. Roy, Health monitoring in an agent-based smart home, in: Proceedings of the International Conference on Aging, Disability and Independence Advancing Technology and Services to Promote Quality of Life, 2003, pp. 3–141.

[3] D.J. Cook, M. Youngblood, S.K. Das, A multi-agent approach to controlling a smart environment, in: Designing smart homes, Springer, 2006, pp. 165–182.

[4] N.C. Krishnan, D.J. Cook, Activity recognition on streaming sensor data, Pervasive Mob. Comput. (2012).

[5] T.V. Kasteren, A. Noulas, G. Englebienne, B. Kröse, B. Kroese, Accurate activity recognition in a home setting, in: Proceedings of the 10th international conference on Ubiquitous computing, ACM, 2008, pp. 1–9.

[6] L. Chen, C.D. Nugent, H. Wang, A knowledge-driven approach to activity recognition in smart homes, IEEE Trans. Knowl. Data Eng. 24 (6) (2012) 961–974.

[7] D. Riboni, C. Bettini, COSAR: hybrid reasoning for context-aware activity recognition, Personal Ubiquitous Comput. 15 (3) (2011) 271–289.

[8] G. Meditskos, I. Kompatsiaris, iKnow: Ontology-driven situational awareness for the recognition of activities of daily living, Pervasive Mob. Comput. 40 (2017) 17–41.

[9] D. Riboni, C. Bettini, Context-aware activity recognition through a combination of ontological and statistical reasoning, in: International Conference on Ubiquitous Intelligence and Computing, Springer, 2009, pp. 39–53.

[10] R. Helaoui, D. Riboni, H. Stuckenschmidt, A probabilistic ontological framework for the recognition of multilevel human activities, in: Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, 2013, pp. 345–354.

[11] Z. Wemlinger, L. Holder, The COSE Ontology: Bringing the semantic web to smart environments, in: International Conference on Smart Homes and Health Telematics (ICOST), Montreal, 2011, pp. 205–209.

[12] D. Bonino, F. Corno, Dogont-ontology modeling for intelligent domotic environments, The Semantic Web-ISWC 2008, 2008.

[13] L. Sommaruga, A. Perri, F. Furfari, DomoML-env: an ontology for Human Home Interaction, in: SWAP, Vol. 166, Citeseer, 2005.

[14] B. Abdulrazak, B. Chikhaoui, C. Gouin-Vallerand, B. Fraikin, A standard ontology for smart spaces, Int. J Web Grid Serv. 6 (3) (2010) 244–268.

[15] A. Sorici, G. Picard, O. Boissier, A. Zimmermann, A. Florea, CONSERT: Applying semantic web technologies to context modeling in ambient intelligence, Comput. Electr. Eng. (2015) http://dx.doi.org/10.1016/j.compeleceng.2015.03.012.

[16] J. Ye, G. Stevenson, S. Dobson, USMART, ACM Trans. Interact. Intell. Syst. 4 (4) (2014) 1–27, http://dx.doi.org/10.1145/2662870.

[17] P. Rashidi, D.J. Cook, Activity knowledge transfer in smart environments, Pervasive Mob. Comput. 7 (3) (2011) 331–343, http://dx.doi.org/10.1016/j.pmcj.2011.02.007.

[18] K.D. Feuz, D.J. Cook, Transfer learning across feature-rich heterogeneous feature spaces via feature-space remapping (FSR), ACM Trans. Intell. Syst. Technol. (TIST) 6 (1) (2015) 3.

[19] K. Feuz, D. Cook, Modeling skewed class distributions by reshaping the concept space, in: AAAI, 2017, pp. 1891–1897.

[20] H. Alemdar, H. Ertan, O.D. Incel, C. Ersoy, ARAS human activity datasets in multiple homes with multiple residents, in: Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2013 7th International Conference on, IEEE, 2013, pp. 232–235.

[21] D.J. Cook, A.S. Crandall, B.L. Thomas, N.C. Krishnan, CASAS : A smart home in a box, IEEE Comput. 46 (6) (2013) 26–33.

[22] D.J. Cook, M. Schmitter-Edgecombe, Assessing the quality of activities in a smart environment, Methods Inf. Med. 48 (5) (2009) 480.

[23] D.J. Cook, Learning setting-generalized activity models for smart spaces, IEEE Intell. Syst. 27 (1) (2010) 3238, http://dx.doi.org/10.1109/mis.2010.112.

[24] L. Klack, C. Möllering, M. Ziefle, T. Schmitz-Rode, Future care floor: A sensitive floor for movement monitoring and fall detection in home environments, in: Wireless Mobile Communication and Healthcare, Springer, 2011, pp. 211–218.

[25] D.J. Cook, M. Schmitter-Edgecombe, P. Dawadi, Analyzing activity behavior and movement in a naturalistic environment using smart home techniques, IEEE J. Biomed. Health Inf. 19 (6) (2015) 1882–1892, http://dx.doi.org/10.1109/JBHI.2015.2461659.

[26] T. Nef, P. Urwyler, M. Büchler, I. Tarnanas, R. Stucki, D. Cazzoli, R. Müri, U. Mosimann, Evaluation of three state-of-the-art classifiers for recognition of activities of daily living from smart home ambient data, Sensors (Switzerland) 15 (5) (2015) 11725–11740, http://dx.doi.org/10.3390/s150511725.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in {P}ython, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[28] P. Flach, M. Kull, Precision-recall-gain curves: PR analysis done right, Adv. Neural Inf. Proc. Syst. 28 1 (2015) 838–846.