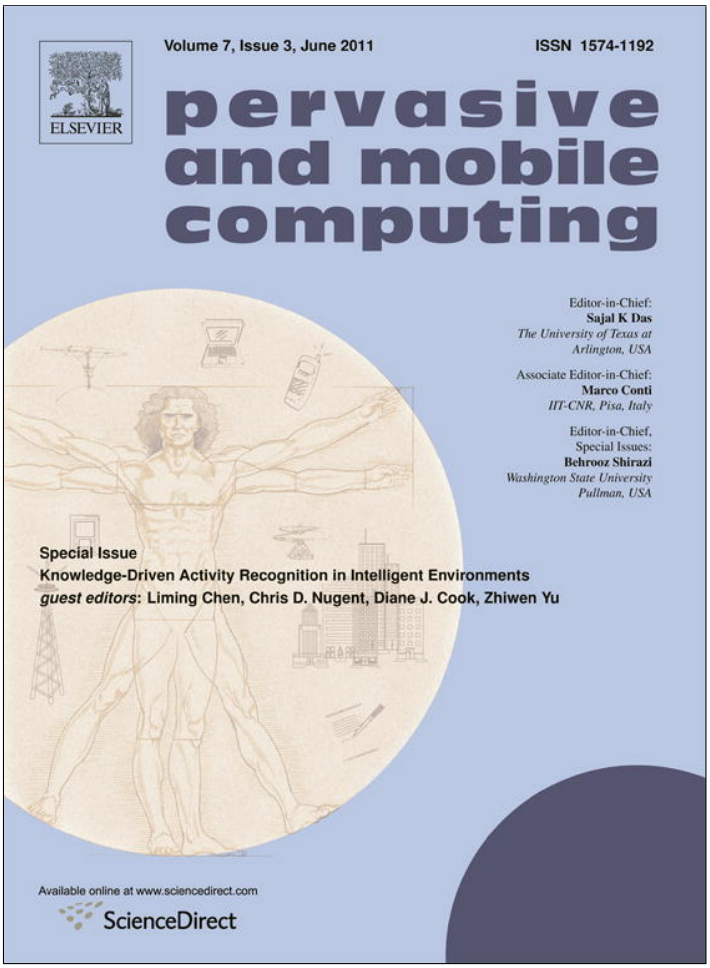


Pervasive and Mobile Computing 7 (2011) 331–343

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

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

Activity knowledge transfer in smart environments

Parisa Rashidi *, Diane J. Cook

Washington State University, Pullman, WA, 99164, United States

ARTICLE INFO

Article history:
Available online 5 March 2011

Keywords:
Smart environments
Activity recognition
Transfer learning

ABSTRACT

Current activity recognition approaches usually ignore knowledge learned in previous smart environments when training the recognition algorithms for a new smart environment. In this paper, we propose a method of transferring the knowledge of learned activities in multiple physical spaces, e.g. homes *A* and *B*, to a new target space, e.g. home *C*. Transferring the knowledge of learned activities to a target space results in reducing the data collection and annotation period, achieving an accelerated learning pace and exploiting the insights from previous settings. We validate our algorithms using data collected from several smart apartments.

1. Introduction

The remarkable recent progress in computing power, networking, and sensors combined with the power of machine learning and data mining techniques have enabled us to create cyber–physical systems that reason intelligently, act autonomously, and respond to the needs of the users in a context-aware manner [1]. Some of the efforts for realizing such cyber–physical systems have been demonstrated in actual smart environment physical testbeds such as the CASAS project [2], the MavHome project [3], the Gator Tech Smart House [4], the iDorm [5], and the Georgia Tech Aware Home [6]. A smart environment typically contains many highly interactive devices and different types of sensors such as motion sensors. The data collected from those sensors can be used in conjunction with data mining and machine learning techniques in order to discover residents' frequent activity patterns [7], predict upcoming events [8], automate interactions with the environment [9], recognize activity patterns [10], and detect abnormal patterns in case of health monitoring or security applications [11]. Recognizing residents' activities allows the smart environment to respond in a context-aware way to residents' needs for achieving more comfort, security, energy efficiency, as well as for monitoring the well being of older adults or people with cognitive and physical impairments [12]. For example, researchers are recognizing that smart environments can be of great value for monitoring and tracking the daily activities of individuals with memory impairments, as the ability to consistently complete Activities of Daily Living (ADLs) [13] is necessary in order to live independently at home. The need for the development of such smart home technologies is underscored by the aging of the population, the cost of formal health care, and the importance that individuals place on remaining independent in their own homes [14].

While smart environments offer many societal benefits, they also introduce new and complex machine learning challenges. A typical home may be equipped with hundreds or thousands of sensors. Because the captured data is rich in structure and voluminous, the learning problem is a challenging one. As a result, each environmental situation has been treated as a separate context in which to perform learning. What can propel research in cyber–physical systems forward is the ability to leverage experience of previous environments in new different environments. However current activity recognition approaches do not exploit the knowledge learned in previous spaces in order to kick start activity recognition in a new space. This results in a delayed installation period in practice due to the need for collecting and annotating huge

* Corresponding author.
E-mail addresses: prashidi@eecs.wsu.edu (P. Rashidi), cook@eecs.wsu.edu (D.J. Cook).

1574-1192/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.pmcj.2011.02.007

amounts of data for each new space. It also leads to redundant computational effort and excessive time investment, and results in ignoring insights gained from previous spaces. In this paper, we propose a method for transferring the knowledge of learned activities from multiple physical smart environments to a new target physical smart environment. In our approach, the layout of the spaces and the residents' schedule can be different. Also the type and number of sensors in two spaces might be different. We validate our algorithms using data collected from six different smart apartments with different layouts and different residents.

2. Related work

Activity recognition has been used in different situations in smart environments, such as for recognizing nurses' activities in hospitals [15], recognizing quality inspection activities during car production [16], or monitoring elderly adults' activities [17]. To discover, recognize and model activities, researchers have devised numerous supervised and unsupervised methods. Naive Bayes classifiers are one of the simplest yet promising supervised methods [18]. Decision trees [19] learn the logical descriptions of the activities in form of rules that are understandable by the user. Markov models [10] and their variations such as hierarchical hidden Markov models (HHMM) [20] recently have been quite popular for recognizing activities. Dynamic Bayesian Networks (DBN) [21] and conditional random fields [22] are yet another flexible framework for recognizing human activity. There also have been a number of unsupervised activity discovery and recognition methods. Gu et al. [23] look for frequent sensor sequences. Pei et al. [24] mine discontinuous activity patterns, and previously we have shown methods to find mixed frequent–periodic activity patterns [2].

The problem with all the above approaches is that they do not exploit the knowledge learned in previous spaces in order to discover and recognize activities in a new space. Therefore, for each new space a huge amount of data needs to be collected and annotated, and the algorithms are trained without benefiting from prior knowledge. Using conventional unsupervised methods such as frequent or periodic data mining methods, the long data collection period and prolonged installation process becomes a problem in practice. Using supervised methods a greater burden is placed on the user of the smart environment, who must annotate sufficient data in order to train the recognition algorithms. Hand labeling from raw sensor data is very time consuming. Data collected in our testbeds required at least one hour of an expert's time to annotate a single day's worth of sensor data. This particularly becomes problematic if we are targeting a deployment in the home of an older adult who may not be able to accurately annotate a large amount of data. Also, learning the model of each environment separately and ignoring what has been learned in other physical settings leads to redundant computational effort, excessive time investment, and loss of beneficial information that can improve the recognition accuracy. Therefore it is beneficial to develop models that can exploit the knowledge of learned activities by employing them in new spaces. Exploiting the transferred knowledge results in reducing the need for data collection, reducing or eliminating the need for data annotation, and accelerating the learning pace. Using multiple sources and fusing their data together can leverage the learning process even more by using a more diverse set of activity models that can help in discovering and recognizing the target activities.

The process of exploiting the knowledge gained in one problem and applying the learned knowledge to a different but related problem is called transfer learning [25,26]. Researchers have studied transfer learning in different computational settings such as reinforcement learning [27], genetic algorithms [28], neural networks [29], Bayesian models [30] and many other methods [31]. Though transfer learning has been vastly studied in the literature [31], it has been applied to activity discovery and recognition in very few cases.

Previously we have shown a method for transferring learned activities from one resident to another to deal with the inter-subject and intra-subject variability [32]. Zhang et al. [33] have developed a model for mapping different types of activities to each other (e.g. sweeping to cleaning) by learning a similarity function via a Web search. Kasteren et al. [34] describe a simple method for transferring the transition probabilities of Markov models for two different spaces. They only transfer the transition probabilities. Other activity features such as the activity's structure and related temporal features are ignored because they assume that the HMM structure is predefined.

In our approach, the activity model includes much more information based on using structural, temporal and spatial features of the activities. Also, unlike the approach of Kasteren et al. [34], we do not manually map the sensor networks. Instead, we learn sensor mappings based on the available data and activity models. It should be noted that in order to exploit the knowledge learned in different spaces, we transfer the activities from multiple physical source spaces to a target physical space. First we use a location based data mining method to find target activities in the target data. Then the activities from both source and target spaces are represented in a canonical form called an "activity template" in order to allow for a more efficient mapping process. Next we use a semi-EM (Expectation Maximization) framework [35] to map source activities from each single source to the target activities. Finally by using an ensemble learning method based on a weighted majority vote [36] and fusing multiple data sources we assign activity labels to the target activities.

3. Model description

Our objective is to develop a method for transferring knowledge in the form of activity models learned in multiple source physical spaces to a target physical space in order to reduce data collection time, reduce or eliminate data annotation time

Table 1
Example sensor data. Here M004, M030 and M015 denote sensor IDs.

Timestamp (t)	Sensor ID (id)	Label (l)
7/17/2009 09:52:25	M004	Personal hygiene
7/17/2009 09:56:55	M030	Personal hygiene
7/17/2009 14:12:20	M015	None

and exploit prior source knowledge in a new target space. We will refer to our method as Multi Home Transfer Learning (MHTL for short). We denote N individual sources as $S_1, \ldots, S_N$ and the single target space as T.

We assume that the physical aspects of the spaces, the number and type of sensors, and also the residents and their schedules can be different. We also do not require all of the activities to exist across all spaces. In this work, the number of available sources (N) is limited and computationally manageable, as reducing the number of sources and source selection is beyond the scope of our paper. We do not discriminate between activities performed by different residents. Multi-resident problems have been studied by several researchers and interested readers can refer to related literature [37]. We also assume that the activities' steps are contiguous, i.e. there is no interrupting event in the middle of a specific activity. The activities are also assumed to be consistent over time, i.e. we assume that their pattern does not change over time. In order to prevent any label mismatch between different environments, we define a set of standard rules for annotating activity data. All the annotations adhere to these standard rules. It is also possible to apply a preprocessing step to achieve a unified labeling.

We hypothesize that we will be able to recognize activities in the target space T using little to no labeled data and using only limited unlabeled data. This assumption turns the nature of the problem into a domain adaptation problem [31]. In other words, labeled data is available in the source domain, but no or few data labels are available in the target domain. This allows us to reduce several weeks or months of data collection and annotation in the target space to only a few days' worth of data collection.

Our ultimate objective is to be able to correctly recognize the activities in the target space. By using our method, labeled target activity data becomes available that can be consumed by conventional learning algorithms to perform activity recognition. The labeled activities also can be used as a bootstrap method for other techniques such as active learning techniques in order to quickly improve the recognition results over time using limited data. In the remainder of this section we describe our notation and also we will provide a high level description of the algorithm.

The input data is a sequence of sensor events e each in the form $e = \langle t, \text{id}, l \rangle$ where t denotes a timestamp, id denotes a sensor ID, and l refers to the activity label, if available. An example showing several sensor events can be seen in Table 1.

As depicted in Table 1, each sensor event can be part of a labeled activity such as the first and second sensor events, or it can have no activity labels such as the third sensor event. Each sensor ID is associated with its room name (e.g., kitchen) which we will refer to as a location tag. We use a standard set of location tags across all different sources. The location tags define a simple ontology based on location of sensors. This facilitates transferring activity patterns between different spaces.

We define an activity as $a = \langle \mathcal{E}, l, t, d, \mathcal{L} \rangle$ where $\mathcal{E}$ is a sequence of n sensor events $\langle e_1, e_2, \ldots, e_n \rangle$, l is its label (if available), t and d are the start time and duration of the activity, and $\mathcal{L}$ represents the set of location tags where a has occurred. Note that the start time and duration in general are represented as a mixture normal distributions, though initially most activities' start time and duration consists only of a single data point, and later during activity consolidation the distribution will be formed. As can be seen from the activity's definition, each activity has structural information in the form of a sensor sequence $\mathcal{E}$, temporal information in the form of t and d, and spatial information in the form of $\mathcal{L}$. These features allow us to convert raw data into an activity model suited for mapping.

We denote the set of activities in each individual source space S_k by $\mathcal{A}_k$. The set of all source activities is denoted by $\mathcal{A}$ where $\mathcal{A}$ is the union of activities from all individual sources, i.e. $\mathcal{A} = \bigcup_k \mathcal{A}_k$. We denote the set of target activities by $\mathcal{A}_T$. The set of all source sensors is denoted by by $\mathcal{R}$, and the set of sensors for S_k is denoted by $\mathcal{R}_k$. The set of target sensors is denoted by $\mathcal{R}_T$.

In order to be able to map activities from the source space to a target space, we need to find a way to map the source sensor network to the target sensor network, as the source sensors can have different locations and properties than the target sensors. Therefore we need to find the mapping $\mathcal{G}(\mathcal{R}_k) = \mathcal{R}_T$. Finding a sensor mapping $\mathcal{G}$ allows us to map a source activity to a target activity based on structural similarity (sensor similarity) and based on the way that the source activity's sensors map to the target activity's sensors. Based on using the sensor mappings $\mathcal{G}$ (the structural mapping) as well as on available temporal and spatial features, we will find the activity mapping function $\mathcal{F}(\mathcal{A}_k) = \mathcal{A}_T$.

To show how well a source activity (or sensor) is mapped to a target activity (sensor), we use mapping probabilities. The mapping probability of an activity $a \in \mathcal{A}_k$ to activity $b \in \mathcal{A}_T$ is reflected in matrix M_k, where $M_k[a, b] \in [0 \ldots 1]$ shows the likelihood that activity a and b have the same label. Similarly, a second matrix $m_k[p, q] \in [0 \ldots 1]$ shows the probability that sensor $p \in \mathcal{R}_k$ maps to sensor $q \in \mathcal{R}_T$ based on their location and their role in activity models. Note that the mappings need not to be one to one, as the number of sensors and also the number of activities can be different in the source and target spaces.

MHTL's activity discovery and knowledge transfer is performed in several stages (see Fig. 1). First we process the labeled data from the source space and we also mine the available unlabeled data from the target space in order to extract the activity models in each space. In the source space, for each individual source S_k we extract the activities $\mathcal{A}_k$ by converting

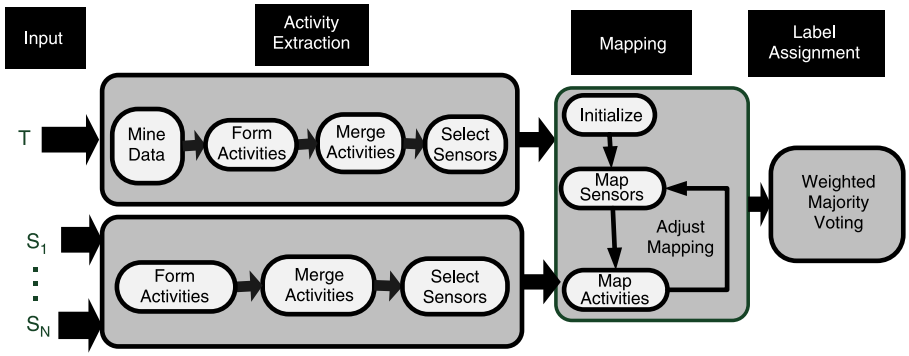


Fig. 1. Main components of MHTL for transferring activities from multiple source spaces to a target space.

Table 2
Some important notations.

Notations	Description	Notation	Description
S_i	Source i	a	Activity
N	Number of sources	d	Activity duration
T	Target	$\mathcal{A}$	All activities
e	Sensor event	$\mathcal{R}$	All sensors
$\mathcal{E}$	Set of sensor events	$\mathcal{F}$	Activity mapping function
t	Start time	$\mathcal{G}$	Sensor mapping function
l	Label	M	Activity mapping probability
$\mathcal{L}$	Activity locations	m	Sensor mapping probability
Υ	Activity similarity	ς	Similarity threshold
r	Time interval	I	Mutual information
Φ	Start time distribution	Γ	Duration distribution

each contiguous sequence of sensor events with the same label to an activity. To reduce the number of activities and to find a canonical mapping, we consolidate similar activities in $\mathcal{A}_k$ together to represent an "activity template". To avoid mapping irrelevant sensors, we use a filter feature selection method based on mutual information [38] to discard the irrelevant sensors for each activity template. In the target space we mine the data to find unlabeled activity patterns based on using location closure. Target activities are then consolidated using an incremental clustering method [39]. If any labeled data is available in the target space, it can be used to refine the target activity models.

In the next step, we map the source activity templates to the target activity templates. First we compute the activity templates' initial mapping probabilities based on structural, temporal and spatial similarities. The sensors' initial mapping probabilities are assigned based on a spatial similarity measure. After initialization, we compute the mapping probabilities in an iterative manner using an Expectation Maximization-like framework [35], which we refer to as a semi-EM process. First, we adjust the sensor mapping probabilities based on the activity mapping probabilities. Next, we adjust the activity mapping probabilities based on the updated sensor mapping probabilities. This continues until no more changes are perceived or until a user-defined number of iterations is reached.

The final step involves assigning labels to target activities. We assign an activity label to each target activity b based on the obtained activity mapping probabilities M. To map activity labels we use an ensemble method based on a weighted majority voting. Each space S_k casts a vote for the label of the target activity b. The voted label is selected the same as the label of a source activity a that maximizes the mapping probability M_k for b. Each vote is weighted by the overall similarity between the source space S_k and the target space T, as will be described later. At the end, the label with the maximum weighted votes is considered as the label of the activity b. Note that in this method all the sources contribute to the label mapping process in order to generate a final activity label for each target activity.

We provide a more detailed description of these steps in Sections 4–6. The important notations are summarized in Table 2.

4. Activity model extraction

The first step of the multi-stage MHTL algorithm is to extract the activity models from input data in both source and target spaces. For each single source space S_k we convert each contiguous sequence of sensor events with the same label to an activity a. This results in finding the set of activities $\mathcal{A}_k$ for each one of the individual source spaces S_k. The start time of the activity is the timestamp of its first sensor event, while its duration is the difference between its last and first timestamps. Due to the prohibitively large number of extracted activities and possible similarity among them, we combine similar activities together as an "activity template". Representing a set of similar activities as an activity template allows for a more efficient canonical mapping from source to target, as only a few activity templates will be mapped from source to target instead of mapping a large number of similar activities with only minor differences. The activity template for a set of

activities is itself an activity, formed by merging activities' sensors, durations, and start times where the merged start times and durations form a mixture normal distribution.

The temporal mixture model allows us to capture and model variations of the same activity that occur at different times. For example, consider the "eating" activity which usually happens three times a day, once in the morning as breakfast, once at noon as lunch, and once at night as dinner. Using a mixture model for the start time we are able to capture all three variations by using a single activity model. The method for obtaining the mixture model is demonstrated in Algorithm 1. The input to Algorithm 1 is the set of all timestamps for an activity a and the time interval granule in hours denoted by r. Using this method allows for a variable number of distributions to be discovered. A similar method is used for obtaining duration distributions where the number of duration distributions equals the number of obtained start time distributions for that activity.

Algorithm 1 The Start Time Mixture Model

procedure FINDMIXTUREMODEL(a, r)
  for all timestamps t belonging to a **do**
    Find $t' \in [1..\frac{24}{r}]$ s.t. $t \in [t' - \frac{r}{2} .. t' + \frac{r}{2}]$ ▷ Find the right interval
    $c[t'] = c[t'] + 1$ ▷ Increase its count
  end for

  $\tilde{c} = \frac{r * \sum c}{24}$
  for all $c[t'] > \tilde{c}$ **do** ▷ If frequency > Average
    Make t' a centroid ▷ Form initial centroids
  end for

  for all timestamps t **do** ▷ Find final centroids
    Assign t to closest centroid
    Recompute centroids
  end for

end procedure

During activity consolidation, all the source activities that have the same label will be merged into one single activity template. Note that as each activity template is itself an activity, we use the terms activity and activity template interchangeably.

The next step after similar activities are consolidated is to perform sensor selection for each activity template a by preserving only relevant sensors. Performing sensor selection on each activity template allows for an even more compact representation and a more accurate mapping, as it allows us to map only the relevant sensors and to avoid mapping the irrelevant sensors as noise. Our sensor selection method is a filter feature selection method based on mutual information [38]. For each activity template a and each sensor s we define their mutual information $I(s, a)$ as in Eq. (1). This value measures their mutual dependence and shows how relevant sensor s is in predicting the activity's label. Here $P(s, a)$ is the joint probability distribution of s and a, while $P(s)$ and $P(a)$ are the marginal probability distributions, all computed from the sensor and activity occurrences in the data. A high mutual information value indicates the sensor is relevant for the activity template. We simply consider sensors with a mutual information above the midpoint (0.5) as relevant, otherwise they will be discarded.

$$I(s, a) = P(s, a) * \log \frac{P(s, a)}{P(s)P(a)}. \tag{1}$$

To find activity patterns in unlabeled target data, we perform a data mining step on the input data. First we partition the input data into activities. A sensor event $e_1 = \langle t_1, \mathrm{id}_1, l_1 \rangle$ and its successor sensor event $e_2 = \langle t_2, \mathrm{id}_2, l_2 \rangle$ are part of the same activity if $L_1 = L_2$, i.e. if both sensors are in the same location. Such a local partitioning allows us to have a baseline for finding individual activities. This approach is based on the intuition that occurrences of the same activity usually happen within the same location (such as preparing meals in the kitchen, grooming in the bathroom, etc.), and more complex activities occurring in different locations can be composed of those basic activities. Notice that as we only have access to limited input data (perhaps a few days or even a few hours), we cannot use conventional activity discovery methods such as frequent or periodic sequence mining methods [40] to find activity patterns in the data. Therefore, exploiting the spatial closure can be a way to overcome this problem.

After partitioning data into initial activities, we consolidate those activities by grouping together similar activities into an activity template. To combine activities together, we use an incremental clustering method [39], such that each activity is assigned to the most similar centroid if their similarity is above threshold ς, and then the centroid is recomputed. Otherwise

the activity forms a separate cluster. The centroid is itself represented as an activity template. At the end all the activities in one cluster are consolidated together and the sensor selection is carried out. For two activities a and b, their similarity $\Upsilon(a, b)$ is defined as in Eq. (2).

$$\Upsilon(a, b) = \Upsilon_t[a, b] + \Upsilon_d[a, b] + \Upsilon_{\mathcal{L}}[a, b] + \Upsilon_S[a, b]. \tag{2}$$

In Eq. (2), Υ_t refers to the start time mapping (if the two activities happen at similar times, e.g., both around noon), Υ_d refers to duration mapping (if the two activities have similar durations), $\Upsilon_{\mathcal{L}}$ refers to location mapping (if the two activities happen in similar locations, e.g., both in the kitchen), and Υ_S refers to structure mapping (if the two activities have a similar structure in terms of sensors). We normalize $\Upsilon(a, b)$ to fall within the range $[0 \ldots 1]$. For simplicity, we have chosen the mappings to have equal effects; however, it is possible to define $\Upsilon(a, b)$ as a weighted average.

As mentioned earlier, the start times are in the form of a mixture normal distribution. We represent the start times in an angular form ϕ measured in radians instead of a linear representation, with means $\Phi = \langle \phi_1 \ldots \phi_n \rangle$. This representation form allows for time differences to be represented correctly (2:00 am will be closer to 12:00 pm than to 5:00 am). The similarity between the two start time distributions is thus calculated using Eq. (3).

$$\Upsilon_t[a, b] = \max_{\substack{\phi_a \in \Phi_a \\ \phi_b \in \Phi_b}} \left(1 - \frac{|\phi_a - \phi_b|}{2\pi}\right). \tag{3}$$

Duration mapping is calculated as in Eq. (4) where durations are in form of a mixture normal distribution with means $\Gamma = \langle \gamma_1 \ldots \gamma_n \rangle$.

$$\Upsilon_d[a, b] = \max_{\substack{\gamma_a \in \Gamma_a \\ \gamma_b \in \Gamma_b}} \left(1 - \frac{|\gamma_b - \gamma_a|}{\max(\gamma_b, \gamma_a)}\right). \tag{4}$$

To compute $\Upsilon_{\mathcal{L}}$ we use Eq. (5) which is the Jaccard similarity coefficient [41] for the sets of locations of the two activities. A similar Jaccard similarity coefficient based on similar sensors is defined for the structure mapping Υ_S in Eq. (6).

$$\Upsilon_{\mathcal{L}}[a, b] = \frac{|\mathcal{L}_a \bigcap \mathcal{L}_b|}{|\mathcal{L}_a \bigcup \mathcal{L}_b|} \tag{5}$$

$$\Upsilon_S[a, b] = \frac{|\mathcal{E}_a \bigcap \mathcal{E}_b|}{|\mathcal{E}_a \bigcup \mathcal{E}_b|}. \tag{6}$$

5. Mapping sensors and activities

The next step after the activity models for the source and target space have been identified is to map the source activity templates to the target activity templates. First we initialize the sensor and activity mapping matrices, m_k and M_k, for each source and target pair (S_k, T). The initial values of the sensor mapping matrix $m_k[p, q]$ for two sensors $p \in \mathcal{R}_k$ and $q \in \mathcal{R}_T$ is defined as 1.0 if they have the same location tag, and as 0 if they have different location tags. The initial value of $M_k[a, b]$ for two activities $a \in \mathcal{A}_k$ and $b \in \mathcal{A}_T$ is obtained based on exploiting related spatial and temporal information and also prior activity label information (if available), as in Eq. (7). Note that in Eq. (7) the first case applies to the few labeled target activities, while for the majority of the target activities the second case is applied.

$$M_k[a, b] = \begin{cases} 1.0 & \text{if } l_a = l_b \\ \Upsilon(a, b) & \text{otherwise.} \end{cases} \tag{7}$$

For computing subsequent mapping probabilities, we use an EM-like framework [35] by estimating the mapping probabilities in an iterative manner. First, the sensor mapping probabilities are computed; and in the next step the activity mapping probabilities are maximized based on the sensor probabilities. Though this model does not exactly reflect an EM algorithm, due to its iterative manner and likelihood estimation in two steps, we refer to it as a semi-EM framework.

To compute sensor mapping probabilities $m_k[p, q]$ for sensors $p \in \mathcal{R}_k$ and $q \in \mathcal{R}_T$, we rely on activities in which p and q appear, as in Eq. (8). The learning rate α refers to how fast we want to converge on the new values, while $m_k^n[p, q]$ and $m_k^{n+1}[p, q]$ refer to the current and updated values of $m_k[p, q]$ in iterations n and $n + 1$, respectively.

$$m_k^{n+1}[p, q] = m_k^n[p, q] - \alpha * \Delta m_k[p, q] \tag{8}$$

$$\Delta m_k[p, q] = m_k^n[p, q] - \frac{1}{|X_p|\,|Y_q|} \sum_{a \in X_p} \sum_{b \in Y_q} M_k[a, b] \tag{9}$$

$$\begin{aligned} X_p &= \{a \in \mathcal{A}_k | p \in \mathcal{E}_a\} \\ Y_q &= \{a \in \mathcal{A}_T | q \in \mathcal{E}_a\}. \end{aligned} \tag{10}$$

In Eq. (9), X_p and Y_q give us all the activities in which sensors p and q appear. This means that those activities which do not include a given sensor will not contribute to that sensor's mapping probability.

In the next step, to adjust the mapping probability between each two activities, we use Eq. (11) to account for the updated sensor mappings. Here $M_k^n[a, b]$ and $M_k^{n+1}[a, b]$ refer to the current and updated values of $M_k[a, b]$ in iteration n and $n + 1$, respectively.

$$M_k^{n+1}[a, b] = M_k^n[a, b] - \alpha * \Delta M_k[a, b] \tag{11}$$

$$\Delta M_k[a, b] = M_k^n[a, b] - \frac{1}{|\mathcal{E}_a|} \sum_{p \in \mathcal{E}_a} \arg\max_{q \in \mathcal{E}_b} \{m_k[p, q]\}. \tag{12}$$

The above procedure for computing sensor mapping and activity mapping probabilities is repeated until no more changes are perceived or until a pre-defined number of iterations is reached. Next, the labels are assigned to the target activities based on the obtained probability mapping matrices.

6. Label assignment

In order to assign labels to the target activities, we use a voting ensemble method [36] based on the activity models $\mathcal{A}_k$ for each space S_k. Combining data from different sources to improve the accuracy and to have access to complimentary information is known as data fusion or as a form of ensemble learning [42]. Ensemble learning is a strategic way to combine multiple models, such as different classifiers or hypotheses to solve a computational problem. In our problem, using multiple sources allows us to fuse data from different sources and to form different activity models, therefore being able to model target activities using knowledge gleamed from a more diverse set of source activities.

In order to be able to successfully apply the ensemble learning technique, an ensemble system needs classifiers whose decision boundaries are adequately different from each other. The most popular method is to use different training datasets to train individual classifiers. The diversity condition of ensemble learning in our problem is achieved by using different training sets from N different physical source spaces, resulting in N different hypotheses. We build a classifier based on each individual hypothesis h_k and then by combining the predicted labels of all classifiers for a certain target activity we are able to make a decision about the activity's final label.

Algorithm 2 The weighted majority vote algorithm for label assignment

procedure AssignLabel($\mathcal{A}$, M, m, b)

  for all S_k **do**
    $l \leftarrow l_a \quad s.t. \quad M_k[a, b] = \max_z(M_k[z, b])$ ▷ Vote

    add l to the set of voted labels

    $Sim(S_k, T) \leftarrow \sum_{a \in \mathcal{A}_k} M_k[a, \mathcal{F}(a)]$ ▷ Find overall similarity

    $W[l] = W[l] + \frac{Sim(S_k, T)}{|\mathcal{A}_k|}$ ▷ Increase label's weight
  end for

  $l_b \leftarrow \max_l W[l]$ ▷ Find the final label

  return l_b ▷ Assigned label is l_b
end procedure

Each hypothesis h_k is constructed based on using the activity templates $\mathcal{A}_k$ for space S_k plus the activity and sensor mapping probabilities M_k and m_k. We represent each hypothesis as $h_k = \{\mathcal{F}(\mathcal{A}_k), \mathcal{G}(\mathcal{R}_k)\}$ where $\mathcal{F}$ and $\mathcal{G}$ denote the activity and sensor mapping functions. For a single space S_k, Eqs. (13)–(15) provide us with the activity mapping function $\mathcal{F}$, sensor mapping function $\mathcal{G}$ and the assigned label l_b for each activity $b \in \mathcal{A}_T$. As can be seen in Eq. (15), the target activity's label is selected to be the same as the label of a source activity $a \in S_k$ that maximizes the mapping probability M_k for a.

$$\mathcal{F}(a) = \max_b(M_k[a, b]) \tag{13}$$

$$\mathcal{G}(p) = \max_q(m_k[p, q]) \tag{14}$$

$$l_b = l_a \quad \text{s.t. } M_k[a, b] = \max_z(M_k[z, b]). \tag{15}$$

In order to combine the assigned labels for each b using different hypotheses, we use the weighted majority voting algorithm as in Algorithm 2. The input of this algorithm are the source activities $\mathcal{A}$, the activity mapping probabilities M,

the sensor mapping probabilities m, and activity $b \in \mathcal{A}_T$. The output of the algorithm is the label of b as l_b. For each source space S_k we find the label of b by using Eq. (15). Each predicted label l is associated with a weight $W[l]$, which is the total similarity between the source S_k and T. The total similarity between S_k and T is calculated as in Eq. (16) by summing over the best mapping from S_k to T for each $a \in S_k$. Obviously a label can be voted for by different hypotheses and its weight will be increased as a result.

$$\mathrm{Sim}(S_k, T) = \sum_{a \in \mathcal{A}_k} M_k[a, \mathcal{F}(a)]. \tag{16}$$

At the end, the label with the greatest number of weighted votes is selected as the label of the activity b. After obtaining the labels of all target activities, we can use the obtained labels to train a conventional activity recognition algorithm. We also can use the labels in conjunction with other techniques such as active learning in order to further improve the results.

7. Experiments

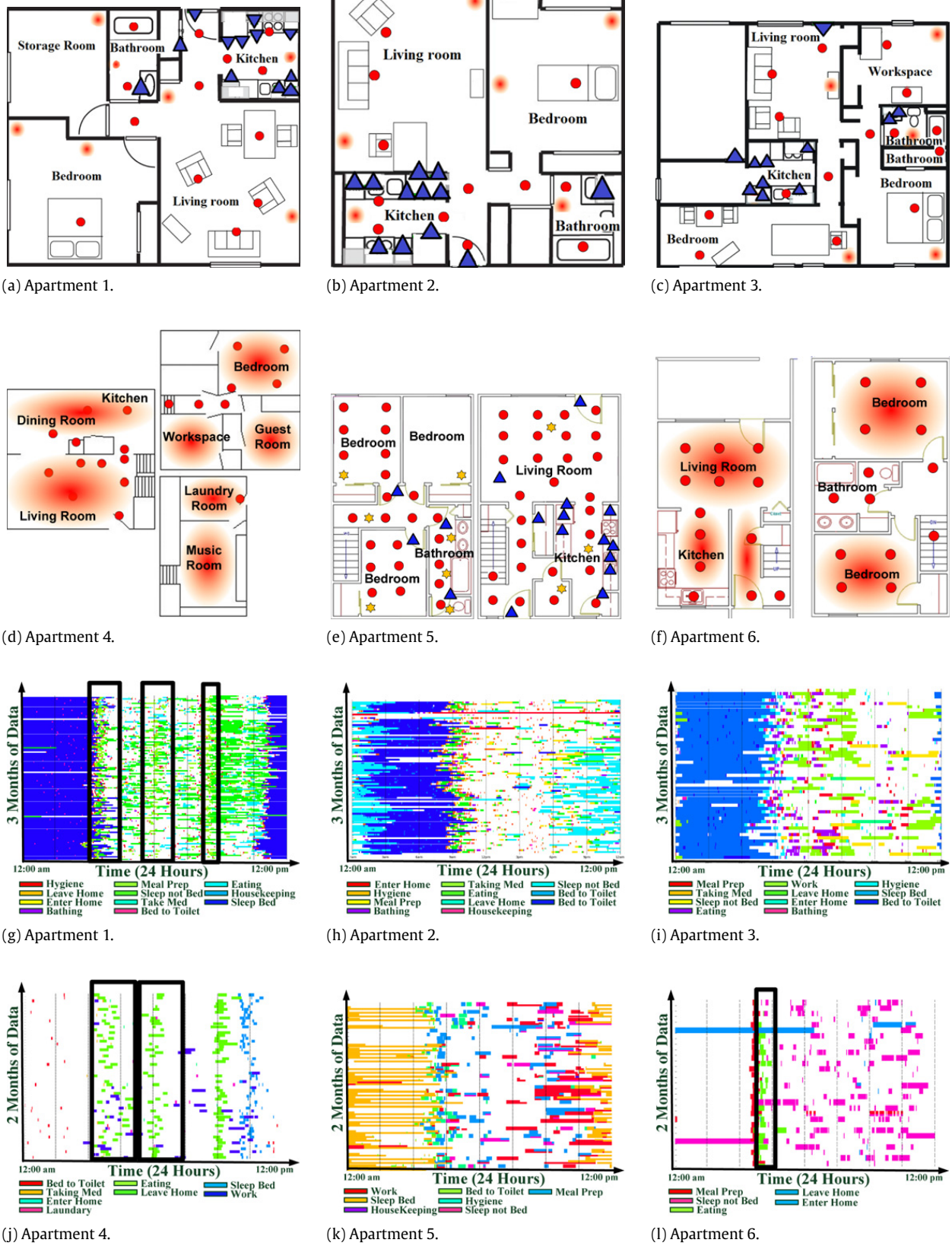
Our approach was implemented and tested as part of the Center for Advanced Studies on Adaptive Systems (CASAS) smart environment project [2] at Washington State University. We evaluated the performance of our MHTL algorithm using the data collected from six different smart apartments. The layout of the apartments, including sensor placement and location tags, are shown in Fig. 2. We will refer to the apartments in Fig. 2(a) through Fig. 2(f) as apartments 1–6. The data was collected during a three month period for apartments 1–3, and during a two month period for apartments 4–6. Each apartment is equipped with motion sensors, and most of the apartments are also equipped with contact sensors which monitor the open/closed status of doors and cabinets. Apartment 5 is also equipped with light sensors and some item sensors to sense the presence of key items. As can be seen in Fig. 2 the apartments have different layouts. For example, apartments 3, 4 and 6 have two bedrooms, while apartments 1 and 2 have one bedroom. In addition, some functional spaces might not be available in all five apartments, such as the workspace, laundry room or the music room. All the sensor data is captured and stored in an SQL database, using a publish/subscribe protocol middleware. To maintain privacy we remove identifying information and encrypt collected data before it is transmitted over the network.

The residents also have quite different schedules, as can be seen from the activity distribution diagrams shown in Fig. 2(g) through Fig. 2(l). For example, in apartment 1 housekeeping is performed each Friday, while in apartment 2 this is performed once a month, and in the third apartment the housekeeping activity is replaced by a work activity. Also the activity level in each apartment is different, as can be seen clearly by comparing activity distribution diagrams for apartments 4 and 6 versus other apartments. The activity level is dependent on the activity level of the residents as well as the number of sensors that monitor the activities. The first three apartments were single resident apartments, while for apartment 4 the residents included a man, a woman, and a cat. Apartments 5 and 6 included two residents. All the data was collected while residents were performing their normal daily activities during a 2–3 month period.

Each of the datasets was annotated with activities of interest for the corresponding resident and apartment. A total of 11 activities were noted for apartments 1–3. Those activities included bathing, bed–toilet transition, eating, entering home, housekeeping (for the third apartment this is replaced by "work"), leaving home, meal preparation, personal hygiene, sleeping in bed, sleeping not in bed (relaxing) and taking medicine. For apartment 4, 7 activities were noted including bed–toilet transition, taking medicine, eating, leaving home, laundry, sleeping in bed and working. Apartment 5 included 7 activities as working, sleeping in bed, bed–toilet transition, personal hygiene, meal preparation, housekeeping, sleeping not in bed (relaxing). Apartment 6 activities included 5 activities as meal preparation, sleeping in bed, eating, leaving home and entering home. As can be seen from activity labels in various spaces, we have defined standard rules for annotating activity data. The annotators are required to adhere to those rules to prevent any label mismatch between different environments. One can expect that in a real world situation either such standardization is enforced, or a preprocessing step is done to achieve a unified labeling.

We ran our algorithm for each one of the apartments as the target space, resulting in six different transfer learning problems. In each setting, all the apartments except for the target apartment were used as the source apartments. In each setting, we used all the available source labeled data, 1–7 days of target unlabeled data, and 0–1 days of target labeled data.

The first step, activity extraction, resulted in a considerable reduction in the number of source activities. In particular 3384, 2602, 1032, 428, 492, and 643 activity instances from apartments 1–6 were represented by as few as 11, 10, 9, 7, 7, and 5 activity templates. The reason that we have obtained less templates than the 11 predefined activities in the second and third apartment is that the "eating" activity was done rather in an erratic way and in different locations, therefore our sensor selection algorithm did not choose any specific sensor for that activity, and as a result the activity was eliminated. The same applied for "taking medicines" in apartment 3. This shows how our algorithm can avoid mapping very irregular activities. It is also possible to obtain a more regular form of such activities by using object sensors such as RFID tags on items. Our results also show how the algorithm condensed the activity instances into a compressed representation, as we approximately obtained the 11 predefined activities for the first three apartments and exactly 7, 7 and 5 activities for the last three apartments. During activity extraction, the number of sensors included for each activity template was reduced from an average of 76.85 sensors to 4.04 sensors, as the algorithm removed the irrelevant sensors and preserved only the relevant sensors. This shows that for each activity a few key sensors can be used to identify the activity, e.g. taking medicine can be identified by the cabinet sensor where the medicines are kept. The algorithm was able to successfully find the mixture



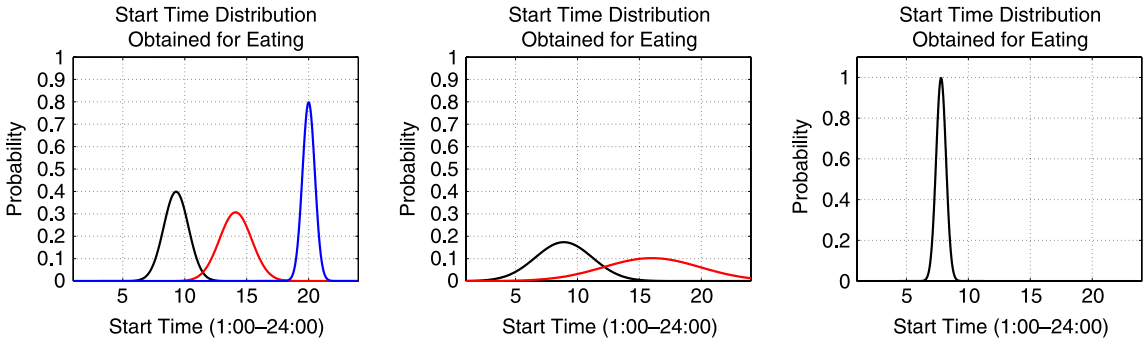
(a) Apartment 1. (b) Apartment 2. (c) Apartment 3.

(d) Apartment 4. (e) Apartment 5. (f) Apartment 6.

(g) Apartment 1. (h) Apartment 2. (i) Apartment 3.

(j) Apartment 4. (k) Apartment 5. (l) Apartment 6.

Fig. 2. Figures (a–e) show sensor map and location tags for each apartment. Circles: motion sensors, triangles: contact sensors, stars: light sensors, hollow-shaped sensors: area motion sensor. Figures (g–l) show residents' activity per a 24 h for 2–3 months of data. The boxes show the approximate boundary of "eating" activity for apartments 1, 4 and 6.



(a) Apartment 1. (b) Apartment 4. (c) Apartment 6.

Fig. 3. Start time's mixture normal distribution for "eating" activity. Here r as the time interval granule was set to 6 h.

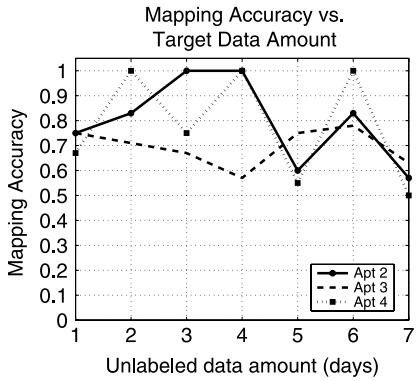
normal distributions for temporal features. Fig. 3 shows the obtained distributions for the "eating" activity in apartments 1, 4, and 6. It can be seen from the distributions that apartment 1 resident has a regular schedule for eating (three times a day). The residents in apartment 4 seem to have a more irregular schedule, usually missing dinner, while apartment 6's residents seem to have a very regular schedule only for breakfast. Comparing the discovered mixture models to the activity distribution diagrams in Fig. 2 confirms the correctness of our results ("eating" activity is surrounded by boxes in Fig. 2(g), (j) and (l)).

In the target space, data was mined to extract the activity templates. For example, using three days of unlabeled target data and no labeled target data, we discovered 8, 7, 7, 5, 5 and 5 activity templates for apartments 1 through 6. The similarity threshold ς in those experiments was set to the midpoint 0.5. The reason that fewer activity templates are discovered compared to the predefined activities is because some similar activities might be merged into one activity, such as relaxing and eating which happen at similar times and similar places. In addition, some activities cannot be easily discovered based only on a few days of data. One example is the housekeeping activity which happens quite rarely compared to other activities; and even if it happens to be in the data, because of its erratic nature and occurring all over the home, it is not very easy to discover.

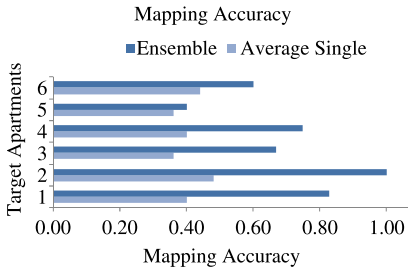
In the next step the source activities were mapped to the target activities. In order to be able to evaluate the mapping accuracy of our algorithm, we embedded the actual labels of target activities in data. This label is not used during training, rather it is only used at the end to verify the correctness of the results. Mapping accuracy is defined as the number of activities in the target space whose transferred label matches the correct expert-supplied label. Fig. 4(a) shows the mapping accuracy for different amounts of unlabeled target data and no labeled target data, in several different settings. Fig. 4(b) shows a comparison between mapping accuracy based on using multiple sources vs. average mapping accuracy using a single source, using 3 days of unlabeled target data. Fig. 4(c) shows the individual single mapping accuracies for apartment 1 versus its ensemble mapping accuracy. It can be seen that on average the ensemble method can beat the single source method. The mapping accuracies vary from space to space, depending on the consistency of activities in target space, as well as the similarity between the source and target spaces, the amount of data available in the source space used to create the activity model, the dimension of the space that has been shifted (new environment, new sensors, or new person). It should be noted that some activities might not be present in all spaces, such as working or housekeeping. The same applies for lack of certain spaces in different apartments, such as the laundry room or the workspace. We noted that transfer between apartments that have a more similar layout and functional structure is more satisfactory.

We tested two of our own activity recognition algorithms on the transferred labeled data. The first algorithm is a nearest neighborhood (1NN) algorithm based on the similarity measure in Eq. (2). The second algorithm represents activities and sensor events with a hidden Markov model and learns the activities using the Viterbi algorithm. The models performed almost the same with the nearest neighborhood algorithm sometimes slightly outperforming HMM due to its use of temporal and spatial features. Though HMM is considered as a temporal method, it should be noted that it considers temporal information in the sense of "order of states", not "start time" or "duration" of states. The HMM does not take into account how much time has been spent in a specific state, or when a transition occurred to another state. Using the embedded labels we define the recognition rate as the percentage of sensor events predicted with the correct label. Fig. 4(d) shows recognition rate of individual activities in apartment 1 using 3 days of unlabeled target data. Fig. 4(e) shows 1NN's recognition rate for apartment 1 as the target apartment using 0 and 1 day of labeled target data. Fig. 4(f) shows 1NN and HMM recognition rates for apartment 1 as the target apartment.

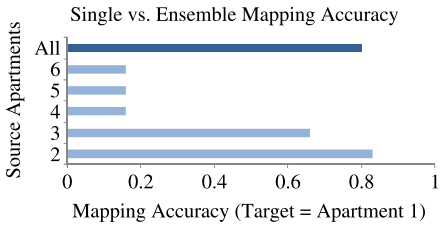
Our results show that despite using little to no labeled target data, and having different layouts, schedules and different activities, both algorithms still achieve a reasonable recognition rate in a target space using data from a source space. It should be noted that in some cases, transferring from a source space to a target space might not provide the best results. This happens when the source and target are not very similar. In such cases, one can apply a source selection method to select the best subset of sources among a set of available sources.



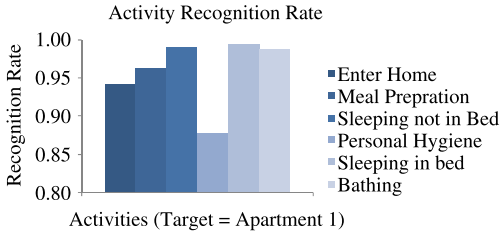
(a) Mapping accuracy in several different settings.



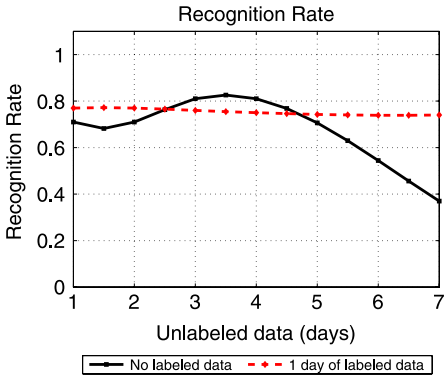
(b) Single vs. ensemble mapping accuracy.



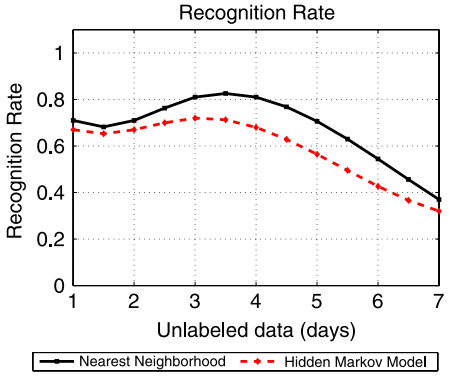
(c) Single vs. ensemble mapping accuracy for apartment 1.



(d) Activity recognition rate for apartment 1.



(e) 1NN's recognition rate using 0 and 1 day of labeled target data.



(f) Recognition rate for 1NN and HMM using no labeled data.

Fig. 4. Mapping accuracies and recognition rates.

8. Summary, limitations, and future work

In this paper we presented a method for transferring the knowledge of learned activities from multiple physical spaces to a new physical space. This allows us to reduce the data collection time, reduce or eliminate the need for data annotation and also exploit the insights from previous spaces. Our experiments show that despite different space layouts and different residents' schedules, we were still able to discover and recognize activities in a target space using no labeled data.

It should be noted that we made a number of simplifying assumptions to deal with the complex nature of activities in the real world. In this work, we ignored the complications that might arise when multiple residents are present. We also made the simplifying assumption that all the patterns are continuous and consist of contiguous sensor events. In reality such assumptions might be violated [43]. Besides, some activities are not always consistent and show a change of pattern over time (e.g. dinner time shifted). This can cause a low recognition rate after some time. We also made the assumption that all the sources are equal for transferring the learned activities. In reality, one of the sources might be more similar to our target space.

One of our future goals is addressing multi-resident issues by using entity detection methods [37]. Transferring sequential patterns that are discontinuous and are disrupted by unrelated sensor events is also part of our future work. As part of our future work, we intend to detect changes in patterns over time. We also intend to devise methods for selecting the best subset of source spaces for training a target space. This allows us to only use the most similar sources and avoid any possible performance degradation due to negative transfer effect. We also want to extend our method to be able to transfer activities across spaces with different functionalities.

Acknowledgements

This research is supported in part by National Science Foundation grant CNS-0852172. The authors also would like to thank Brian Thomas for developing the visualizer software and making available the activity distribution diagrams.

References

[1] D. Cook, S. Das, Smart Environments: Technology, Protocols and Applications, in: Wiley Series on Parallel and Distributed Computing, Wiley-Interscience, 2004.
[2] P. Rashidi, D.J. Cook, the resident in the loop: adapting the smart home to the user, IEEE Transactions on Systems, Man & Cybernetics Journal, Part A (Systems & Humans) 39 (5) (2009) 949–959.
[3] D. Cook, M. Youngblood, I. Heierman, O. Edwin, K. Gopalratnam, S. Rao, A. Litvin, F. Khawaja, Mavhome: an agent-based smart home, in: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003, pp. 521–524.
[4] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, E. Jansen, The gator tech smart house: a programmable pervasive space, Computer 38 (3) (2005) 50–60.
[5] F. Doctor, H. Hagras, V. Callaghan, A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments, IEEE Transactions on Systems, Man & Cybernetics Journal, Part A (Systems & Humans) 35 (1) (2005) 55–65.
[6] G. Abowd, E. Mynatt, Smart Environments: Technology, Protocols, and Applications, in: Designing for the human experience in smart environments, Wiley, 2004, pp. 153–174. (Chapter).
[7] E.O. Heierman III, D.J. Cook, Improving home automation by discovering regularly occurring device usage patterns, in: ICDM'03: Proceedings of the Third IEEE International Conference on Data Mining, 2003, p. 537.
[8] K. Gopalratnam, D.J. Cook, Online sequential prediction via incremental parsing: the active lezi algorithm, IEEE Intelligent Systems 22 (1) (2007) 52–58.
[9] P. Rashidi, D.J. Cook, Adapting to resident preferences in smart environments, in: AAAI Workshop on Preference Handling, 2008, pp. 78–84.
[10] L. Liao, D. Fox, H. Kautz, Location-based activity recognition using relational Markov networks, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2005, pp. 773–778.
[11] T. Yiping, J. Shunjing, Y. Zhongyuan, Y. Sisi, Detection elder abnormal activities by using omni-directional vision sensor: activity data collection and modeling, in: SICE-ICASE, 2006, International Joint Conference, 2006, pp. 3850–3853.
[12] C. Wren, E. Munguia-Tapia, Toward scalable activity recognition for sensor networks, in: Proceedings of the Workshop on Location and Context-Awareness, 2006, pp. 218–235.
[13] B. Reisberg, S. Finkel, J. Overall, N. Schmidt-Gollas, S. Kanowski, H. Lehfeld, F. Hulla, S.G. Sclan, H.-U. Wilms, K. Heininger, I. Hindmarch, M. Stemmler, L. Poon, A. Kluger, C. Cooler, M. Bergener, L. Hugonot-Diener, P.H. Robert, H. Erzigkeit, The Alzheimer's disease activities of daily living international scale (adl-is), International Psychogeriatrics 13 (02) (2001) 163–181.
[14] V. Rialle, C. Ollivet, C. Guigui, C. Hervé, What do family caregivers of Alzheimer's disease patients desire in smart home technologies? Contrasted results of a wide survey, Methods of Information in Medicine 47 (2008) 63–69.
[15] D. Sánchez, M. Tentori, J. Favela, Activity recognition for the smart hospital, IEEE Intelligent Systems 23 (2) (2008) 50–57.
[16] G. Ogris, T. Stiefmeier, P. Lukowicz, G. Troster, Using a complex multi-modal on-body sensor system for activity spotting, in: ISWC'08: Proceedings of the 2008 12th IEEE International Symposium on Wearable Computers, 2008, pp. 55–62.
[17] D. Cook, M. Schmitter-Edgecombe, Assessing the quality of activities in a smart environment, Methods of Information in Medicine 48 (05) (2009) 480–485.
[18] T. van Kasteren, B. Krose, Bayesian activity recognition in residence for elders, in: 3rd International Conference on Intelligent Environments, 2007, pp. 209–212.
[19] U. Maurer, A. Smailagic, D.P. Siewiorek, M. Deisher, Activity recognition and monitoring using multiple sensors on different body positions, in: BSN'06: Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks, 2006, pp. 113–116.
[20] S. Luhr, H. Bui, S. Venkatesh, G. West, Recognition of human activity through hierarchical stochastic learning, in: Pervasive Computing and Communications, 2003, PerCom 2003, Proceedings of the First IEEE International Conference on, 2003, pp. 416–422.
[21] T. Inomata, F. Naya, N. Kuwahara, F. Hattori, K. Kogure, Activity recognition from interactions with objects using dynamic Bayesian network, in: Casemans'09: Proceedings of the 3rd ACM International Workshop on Context-Awareness for Self-Managing Systems, 2009, pp. 39–42.
[22] M. Philipose, K. Fishkin, M. Perkowitz, D. Patterson, D. Fox, H. Kautz, D. Hahnel, Inferring activities from interactions with objects, IEEE Pervasive Computing 3 (4) (2004) 50–57.
[23] T. Gu, Z. Wu, X. Tao, H. Pung, J. Lu, Epsicar: an emerging patterns based approach to sequential, interleaved and concurrent activity recognition, in: Proceedings of the IEEE International Conference on Pervasive Computing and Communication, 2009.
[24] J. Pei, J. Han, W. Wang, Constraint-based sequential pattern mining: the pattern-growth methods, Journal of Intelligent Information Systems 28 (2) (2007) 133–160.
[25] R. Caruana, Multitask learning, Machine Learning 28 (1) (1997) 41–75.
[26] R. Raina, A.Y. Ng, D. Koller, Constructing informative priors using transfer learning, in: ICML'06: Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 713–720.
[27] M. Asadi, M. Huber, Effective control knowledge transfer through learning skill and representation hierarchies, in: IJCAI'07: Proceedings of the 20th International Joint Conference on Artificial Intelligence, 2007, pp. 2054–2059.
[28] M.E. Taylor, P. Stone, Y. Liu, Transfer learning via inter-task mappings for temporal difference learning, Journal of Machine Learning Research 8 (2007) 2125–2167.
[29] S. Thrun, Is learning the *n*th thing any easier than learning the first?, in: Advances in Neural Information Processing Systems, vol. 8, 1996, pp. 640–646.
[30] D.M. Roy, L.P. Kaelbling, Efficient Bayesian task-level transfer learning, in: IJCAI'07: Proceedings of the 20th International Joint Conference on Artifical Intelligence, 2007, pp. 2599–2604.
[31] S.J. Pan, Q. Yang, A survey on transfer learning, Tech. Rep. HKUST-CS08-08, Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China, November 2008.
[32] P. Rashidi, D.J. Cook, Transferring learned activities in smart environments, in: 5th International Conference on Intelligent Environments, in: Ambient Intelligence and Smart Environments, vol. 2, 2009, pp. 185–192.

[33] V.W. Zheng, D.H. Hu, Q. Yang, Cross-domain activity recognition, in: Ubicomp'09: Proceedings of the 11th International Conference on Ubiquitous Computing, 2009, pp. 61–70.
[34] T. van Kasteren, G. Englebienne, B. Krose, Recognizing activities in multiple contexts using transfer learning, in: AAAI AI in Eldercare Symposium, 2008.
[35] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journals of the Royal Statistical Society 39 (1) (1977) 1–38.
[36] T.G. Dietterich, Ensemble methods in machine learning, in: MCS'00: Proceedings of the First International Workshop on Multiple Classifier Systems, 2000, pp. 1–15.
[37] A. Crandall, D. Cook, Attributing events to individuals in multi-inhabitant environments, in: Intelligent Environments, 2008 IET 4th International Conference on, 2008, pp. 1 –8.
[38] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.
[39] F. Can, Incremental clustering for dynamic information processing, ACM Transactions on Information Systems 11 (2) (1993) 143–164.
[40] P. Rashidi, D.J. Cook, An adaptive sensor mining framework for pervasive computing applications, in: International Workshop on Knowledge Discovery from Sensor Data, Sensor-KDD 2008, 2008, pp. 41–49.
[41] P.-N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Adison Wesley, 2005.
[42] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, G.E. Hinton, Adaptive mixtures of local experts, Neural Computation 3 (1) (1991) 79–87.
[43] P. Rashidi, D.J. Cook, L.B. Holder, M. Schmitter-Edgecombe, Discovering activities to recognize and track in a smart environment, IEEE Transactions on Knowledge and Data Engineering 23 (2011) 527–539. http://doi.ieeecomputersociety.org/10.1109/TKDE.2010.148.