Activity Recognition Based on Home to Home Transfer Learning

Parisa Rashidi and **Diane J. Cook**
School of Electrical Engineering and Computer Science,
Washington State University,
Pullman, WA, 99163
{ prashidi,cook}@eecs.wsu.edu

Abstract

Activity recognition plays an important role in many areas such as smart environments by offering unprecedented opportunities for assisted living, automation, security and energy efficiency. It's also an essential component for planning and plan recognition in smart environments. One challenge of activity recognition is the need for collecting and annotating huge amounts of data for each new physical setting in order to be able to carry out the conventional activity discovery and recognition algorithms. This extensive initial phase of data collection and annotation results in a prolonged installation process and excessive time investment for each new space. In this paper we propose a new method of transferring learned knowledge of activities to a new physical space in order to leverage the learning process in the new environment. Our method called "Home to Home Transfer Learning" (HHTL) is based on using a semi EM framework and modeling activities using structural, temporal and spatial features. This method allows us to avoid the tedious task of collecting and labeling huge amounts of data in the target space, and allows for a more accelerated and more scalable deployment cycle in the real world. It also allows us to exploit the insights learned in previous spaces. To validate our algorithms, we use the data collected in several smart apartments with different physical layouts.

Introduction

With remarkable recent progress in sensor technology and machine learning techniques, activity recognition is becoming an integral part of many pervasive computing systems and smart environments. A smart environment typically contains many embedded sensors such as motion sensors that provide opportunities for assisted living and health monitoring, automation, and energy efficiency based on activity recognition and inferring users behaviors from the observations (Cook and Das 2004). Activity recognition is an essential component for planning and plan recognition in smart environments. Typically in a smart environment, first the activities are recognized, and then based on the activity data the plans are recognized and are adjusted. There are many ways in which smart homes rely on activity recognition for planning. Some examples include providing guidance or reminders to a resident during a task, performing actions to help the resident complete a task, or identifying emergencies if the resident is not doing what s/he is supposed to do or is doing something the wrong way (Simpson et al. 2006). Activity recognition and planning can be especially useful in the area of assisted living by monitoring the daily activities of elderly adults with memory deficiencies and helping them via timely prompts and task completion. Typically a set of day to day activities called Activities of Daily Living (ADL) are monitored, including eating, bathing, etc. The completion of ADLs indicates the ability of the individual for an independent life (Rialle et al. 2008).

Besides assisted living and health monitoring applications, activity recognition in smart environments can be used in a variety of other different situations, and in general it can be used to respond to residents' needs in a context-aware manner (Wren and Munguia-Tapia 2006). In this paper we focus on the activity recognition component of smart environments.

Some of the recent smart environment efforts have been demonstrated in actual physical testbeds such as the CASAS project (Rashidi and Cook 2009a), the MavHome project (Cook et al. 2003), the Gator Tech Smart House (Helal et al. 2005), the iDorm (Doctor, Hagras, and Callaghan 2005), and the Georgia Tech Aware Home (Abowd and Mynatt 2004). There also have been a number of stand-alone or ambient cognitive orthotics systems. Early cognitive orthotics systems were based on simple reminders such as the PEAT system which relies on automated planning to provide visible and audible clues about plan execution (Levinson 1997). The COACH reminder system recognizes the hand washing activity and provides a step by step plan in form of useful prompts. Other projects such as the CASAS project (Rashidi and Cook 2009a), Assisted Cognition Project (Dieter et al. 2002), and SOPRANO (Sixsmith et al. 2009) aim at developing cognitive orthotic systems for people with Alzheimer's disease by using ubiquitous sensors to monitor the performance of routine tasks and providing prompts when the user gets stuck or is confused.

In all above projects, activity recognition plays an important role. There have been a number of supervised methods for recognizing activities, such as naive Bayes (Brdiczka, Maisonnasse, and Reignier 2005), decision trees (Maurer et al. 2006), Markov models (Liao, Fox, and Kautz 2005), dynamic Bayes networks (Inomata et al. 2009), and conditional random fields (Philipose et al. 2004). The unsupervised activity discovery and recognition methods include frequent sensor mining methods (Gu et al. 2009), discontinuous activity pattern discovery methods (Pei, Han, and Wang 2007), and methods for finding mixed frequent-periodic activity patterns (Rashidi and Cook 2009a). None of these approaches address the issue of transferring the learned knowledge of activities to new contexts in order to make the systems more scalable. Instead they learn the model of each environment separately. 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. Our testbeds have 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. Also by ignoring what has been learned in other physical settings, we are faced with a redundant computational effort and excessive time investment to learn a new model. Therefore it is beneficial to develop models that can exploit the knowledge of learned activities by employing it in new spaces, thereby reducing or eliminating the need for data annotation, reducing the data collection time, and achieving an accelerated learning pace.

Exploiting the knowledge gained in one problem and applying it to a different but related problem is called transfer learning (Raina, Ng, and Koller 2006), (Caruana 1997). Researchers have studied transfer learning in different computational settings such as reinforcement learning (Asadi and Huber 2007), genetic algorithms (Taylor, Stone, and Liu 2007), neural networks (Thrun 1996), Bayesian models (Roy and Kaelbling 2007) and many other methods (Pan and Yang 2008). As a hallmark of human intelligence, transfer learning has been vastly studied in the literature (Pan and Yang 2008), but it has been applied to activity recognition in very few cases.

In previous works, it has been shown how to transfer the activity models learned for one person to another in the same physical setting (Rashidi and Cook 2009b). Zhang et al. (Zheng, Hu, and Yang 2009) have developed a model that maps different types of activities to each other (e.g. sweeping to cleaning) by learning a similarity function via a Web search. Our goal is to transfer activities between different physical spaces where the physical aspects of the spaces, the residents and the sensor can be different. Kasteren et. al (van Kasteren, Englebienne, and Krose 2008) describe a simple method for transferring the transitional probabilities of Markov models for two different spaces. By reducing activity models to two simple HMMs, their work does not address how activities in a target context can be found using knowledge from the source space except for the transitional probabilities, and they ignore most of the activities' important features such as the activity's structure and related temporal features. They also manually map the sensors from source to target space, which is done automatically in our approach. More importantly, they assume that the structure of HMMs is given and pre-defined, but in our model we make no assumption about the structure of the activities in the target space. The activity model in our work is much more sophisticated, and is based on using structural, temporal and spatial features of activities.

The remainder of this paper is organized as follows. First we describe our two stage approach in more detail, wherein the first stage of our algorithm mines target data and extracts activity models from both spaces, and the second stage maps activity models from source to target environment using a semi EM framework. Next we present the result of our experiments on the data obtained from three smart apartments, and finally we will end the paper with our conclusions and discussion of the future works.

Model Description

Our objective is to develop a method that can transfer learned activities across different physical spaces in order to label and recognize the unlabeled activities in a target space. We assume that labeled activity data is available in the source space S, and the objective is to use such a knowledge to learn the activity "labels" in a target space 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. Similarly, we do not require all of the same activities to exist across all spaces. We assume that the nature of the problem is "inductive transfer learning" or "self taught" (Pan and Yang 2008), i.e. we have labeled data in the source domain, and none 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 of data collection. Our ultimate objective is to be able to correctly recognize 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, or can be used as a baseline for other techniques such as active learning techniques.

The input data is a sequence of sensor events e in the form of $e = \langle t, s, l \rangle$ where t denotes a timestamp, s denotes a sensor ID, and l is 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 is tagged with its associated room name (e.g. kitchen) which we will refer to as a location tag L. A standard set of location tags is used across all different sources. 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 $\langle e_1, e_2, ..e_n \rangle$, l is its label (if available), t and d are the start time and duration distributions, 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 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 sensor sequence $\mathcal{E}$, temporal information in the form of t and

Timestamp (t)	Sensor ID (s)	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

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

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.

In our notation we consider $\mathcal{A}_S$ as the set of source activities, $\mathcal{A}_T$ as the set of target activities, $\mathcal{S}_s$ as the set of source sensors, and $\mathcal{S}_T$ as the set of target sensors. In order to be able to map activities, we need to find a way to map the source sensor network to the target sensor network i.e. we're looking for the mapping $\acute{\mathcal{F}}(\mathcal{S}_S) = \mathcal{S}_T$, as the source sensors will have different locations and properties than the target sensors. Based on using activity features and also $\acute{\mathcal{F}}$, we will find the activity mapping function $\mathcal{F}(\mathcal{A}_S) = \mathcal{A}_T$.

The extent to which activity $a_i \in \mathcal{A}_S$ maps to activity $a_j \in \mathcal{A}_T$ is reflected in matrix M, where $M[i, j] \in [0..1]$ shows the probability that activity a_i and a_j have the same label. Similarly, a second matrix $m[p, q] \in [0..1]$ shows the probability that sensor $s_p \in \mathcal{S}_S$ maps to sensor $s_q \in \mathcal{S}_T$ based on their location and their role in activity models. Note that the mappings need not to be one to one, due to the differences in the number of sensors and number of activities in the source and target spaces. It's also possible that the mapped sensors are of different types.

Our model performs activity transfer from a source space to a target space in several stages (Figure 1). First, labeled data from the source space and unlabeled data from the target space are processed in order to extract the activity models in each space. In the source space, each contiguous sequence of sensor events with the same label is converted to an activity. To reduce the number of activities and find a canonical mapping, similar activities are consolidated together to represent a "activity template". To avoid mapping irrelevant sensors, a filter feature selection method based on mutual information(Guyon and Elisseeff 2003) is used to remove the irrelevant sensors for each activity template. In the target space the data is mined to find unlabeled activity patterns. Activities are then consolidated using an incremental clustering method (Can 1993). If any labeled data is available in the target space, it can be used to refine the target activity models.

In the next step, source activity models are mapped to the target activity models. First the activities' initial mapping probabilities are computed based on structural, temporal and spatial similarities. The sensors' initial mapping probabilities are assigned based on a simple spatial similarity measure. After initialization, the algorithm starts the semi-EM framework in an iterative manner. First, the sensor mapping probabilities are adjusted based on the activity mapping probabilities, next the activity mapping probabilities are adjusted 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. A target activity's label is chosen to be the same as the source activity's label that maximizes the mapping probability. We will provide a more detailed description of each of the above steps in the following subsections.

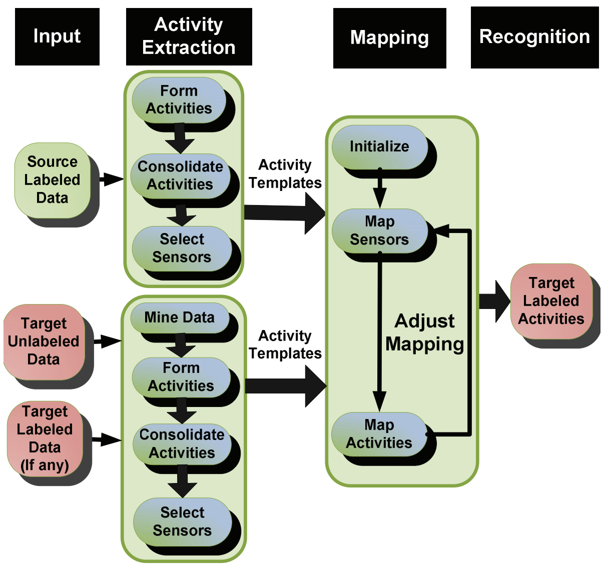


Figure 1: Main components of HHTL for transferring activities from a source space to a target space.

Activity Extraction

The first step of the HHTL algorithm is to extract the activity models from the input data. In the source space, each contiguous sequence of sensor events with the same label is converted to an activity. 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, it is necessary to combine similar activities together as an "activity template". This allows for a more efficient canonical mapping from source to target instead of mapping a large number of activities with only minor differences. The activity template for a set of activities is 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.

In the source space, all the source activities that have the same label will be consolidated into one single activity template. Note that as the activity template is an activity itself, we use the terms activity and activity template interchangeably. After similar activities are consolidated together, we need to perform sensor selection for each activity template by preserving only relevant sensors. This allows us to map only relevant sensors and avoid mapping irrelevant sensors as noise. Our sensor selection method is a filter feature selection method based on mutual information (Guyon and Elisseeff 2003). For each activity template a and each sensor s we define their mutual information $MI(s, a)$ as in Equation 1. It measures their mutual dependence and shows how relevant is sensor s 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, we simply consider sensors with a mutual information above the midpoint (0.5) as relevant, otherwise they will be discarded.

$$MI(s, a) = P(s, a) * \log \frac{P(s, a)}{P(s)P(a)} \quad (1)$$

To extract meaningful structure from unlabeled target data, we perform data mining on the input data. First we partition the input data into activities. A sensor event $e_1 = \langle t_1, s_1, l_1 \rangle$ and a successor sensor event $e_2 = \langle t_2, s_2, l_2 \rangle$ are part of the same activity if $L_{s_1} = L_{s_2}$, 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 are usually within the same location (such as preparing meal 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 (Rashidi and Cook 2008) to find activity patterns in the data. Therefore exploiting the spatial closure can be a way to overcome this problem. After partitioning data into the 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 (Can 1993), 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 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_i and a_j, their similarity $\Upsilon(i, j)$ is defined as in Equation 2.

$$\Upsilon(i, j) = M_t[i, j] + M_d[i, j] + M_{\mathcal{L}}[i, j] + M_S[i, j] \quad (2)$$

In above equation, M_t refers to start time mapping (if the two activities happen at similar times, e.g. both around noon), M_d refers to duration mapping (if the two activities have similar durations), $M_{\mathcal{L}}$ refers to location mapping (if the two activities happen in similar locations, e.g. both in the kitchen), and M_S refers to structure mapping (if the two activities have similar structure in terms of sensors). We normalize $\Upsilon(i, j)$ to fall within the range $[0..1]$. For simplicity, we have chosen the mappings to have equal effects, however it's possible to define $\Upsilon(i, j)$ as a weighted average.

As mentioned, the start times are in form of a mixture normal distribution with means $\Theta = \langle \theta_1..\theta_k \rangle$. We represent start time θ in an angular form Φ measured in radians instead of a linear representation. This allows for time differences to be represented correctly (2:00 am will be closer to 12:00 am rather than 5:00 am). Then the similarity between the two start time distributions will be as in Equation 3.

$$M_t[i, j] = \max_{\substack{\theta_1 \in \Theta_i \\ \theta_2 \in \Theta_j}} (1 - \frac{|\Phi_{\theta_2} - \Phi_{\theta_1}|}{2\pi}) \quad (3)$$

Duration mapping is calculated as in Equation 4 where durations are in form of a mixture normal distribution with means $\Gamma = \langle \gamma_1..\gamma_k \rangle$.

$$M_d[i, j] = \max_{\substack{\gamma_1 \in \Gamma_i \\ \gamma_2 \in \Gamma_j}} (1 - \frac{|\gamma_2 - \gamma_1|}{\max(\gamma_2, \gamma_1)}) \quad (4)$$

To compute $M_{\mathcal{L}}$ we use Equation 5 which is the Jaccard similarity coefficient (Tan, Steinbach, and Kumar 2005) for the sets of locations of the two activities. A similar Jaccard similarity coefficient based on similar sensors is defined for the structure mapping M_S in Equation 6.

$$M_{\mathcal{L}}[i, j] = \frac{| \mathcal{L}_i \bigcap \mathcal{L}_j |}{| \mathcal{L}_i \bigcup \mathcal{L}_j |} \quad (5)$$

$$M_S[i, j] = \frac{| \mathcal{E}_i \bigcap \mathcal{E}_j |}{| \mathcal{E}_i \bigcup \mathcal{E}_j |} \quad (6)$$

Mapping Sensors and Activities

After the activity models for the source and target space have been identified, the source activity templates are mapped to the target activity template. The first step is initializing the sensor and activity mapping matrixes, m and M. The initial values of the sensor mapping matrix $m[p, q]$ for two sensors s_p and s_q is defined as 1.0 if they have the same location tag, and as 0 if they have different locations tags. The initial value of $M[i, j]$ for two activities $a_i \in \mathcal{A}_S$ and $a_j \in \mathcal{A}_T$ is obtained based on exploiting related spatial and temporal information and also prior activity label information (if available), as in Equation 7. Note that in Equation 7 the first case applies to the few labeled target activities, while for the majority of the target activities the second case is applied.

$$M[i, j] = \begin{cases} 1.0 & if \quad l_i = l_j \\ \Upsilon(i, j) & otherwise \end{cases} \quad (7)$$

For computing subsequent mapping probabilities, we use an Expectation Maximization (EM) like framework (Dempster, Laird, and Rubin 1977) 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 doesn't exactly reflect an EM algorithm, however due to its iterative manner and likelihood estimation in two steps, we call it a semi-EM framework.

To compute sensor mapping probabilities $m[p,q]$ for sensors $s_p \in \mathcal{S}_s$ and $s_q \in \mathcal{S}_T$, we rely on activities in which s_p and s_q appear in, as in Equation 8. The learning rate α refers to how fast we want to converge on the new values, while $m^n[p,q]$ and $m^{n+1}[p,q]$ refer to the current and updated values of $m[p,q]$ in iteration n and $n+1$, respectively.

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

$$\Delta m[p,q] = m^n[p,q] - \frac{1}{|X_p||Y_q|} \sum_{a_i \in X_p} \sum_{a_j \in Y_q} M[i,j] \tag{9}$$

$$\begin{aligned} X_p &= \{a_i \in \mathcal{A}_S | s_p \in \mathcal{E}_i\} \\ Y_q &= \{a_j \in \mathcal{A}_T | s_q \in \mathcal{E}_j\} \end{aligned} \tag{10}$$

In Equation 9, X_p and Y_q for sensor p and q give us all the activities in which the sensors 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 Equation 11 to account for the updated sensor mappings. Here $M^n[i,j]$ and $M^{n+1}[i,j]$ refer to the current and updated values of $M[i,j]$ in iteration n and $n+1$, respectively.

$$M^{n+1}[i,j] = M^n[i,j] - \alpha * \Delta M[i,j] \tag{11}$$

$$\Delta M[i,j] = M^n[i,j] - \frac{1}{|\mathcal{E}_i|} \sum_{s_p \in \mathcal{E}_i} \max_{\substack{q \\ s_q \in \mathcal{E}_j}} m[p,q] \tag{12}$$

The above procedure for computing sensor mapping probability and activity mapping probability 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. To assign labels to the target activities and also find sensor mappings, we use Equation 13, 14, and 15 which provide us with the mapping functions $\mathcal{F}$ and $\acute{\mathcal{F}}$ as well as the assigned label l_{a_j} for an activity $a_j \in \mathcal{A}_T$.

$$\mathcal{F}(a_i) = \max_{a_j}(M[i,j]) \tag{13}$$

$$\acute{\mathcal{F}}(s_p) = \max_{s_q}(m[p,q]) \tag{14}$$

$$l_{a_j} = l_{a_i} \quad s.t. \quad M[i,j] = \max_k(M[k,j]) \tag{15}$$

Experiments

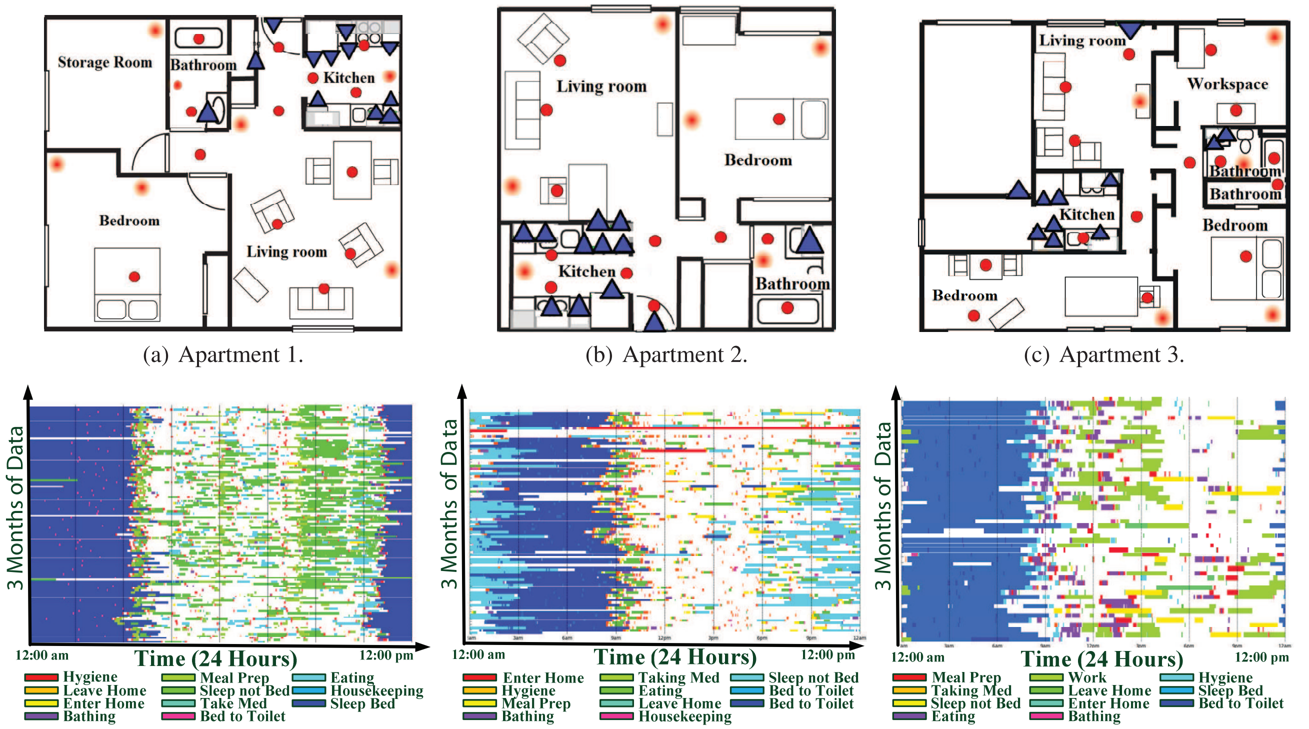
We evaluated the performance of HHTL using data collected from three different apartments during a 3 month period. Each apartment is equipped with motion sensors and contact sensors which monitor the open/closed status of doors and cabinets. The layout of the apartments including sensor placement and location tags are shown in Figure 2(a), Figure 2(b) and Figure 2(c). The apartments have different layouts: the third apartment has 2 bedrooms, 2 bathrooms and a workspace, while the first and second apartments have 1 bedroom and 1 bathroom. All the sensor data is captured and stored in a 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 have quite different schedules, as can be seen in activity distributions in Figures 2(d), 2(e) and 2(f). For example, in the first apartment housekeeping is performed each Friday, while in the second apartment it's performed once a month, and in the third apartment the housekeeping activity is replaced by work activity. Each of the three datasets was annotated with activities of interest for the corresponding resident and apartment. A total of 11 activities were noted in each case which include bathing, bed-toilet transition, eating, enter home, housekeeping (for the third apartment this is replaced by "work"), leave home, meal preparation, personal hygiene, sleeping in bed, sleeping not in bed (relaxing) and taking medicine.

We ran our algorithm on each pair of apartments, resulting in six different transfer learning problems. In each setting, we used 3 months of source labeled data, 1 to 14 days of target unlabeled data, and 0 to 1 days of target labeled data.

The first step, activity extraction, resulted in a considerable reduction in the number of activities. In particular 3384, 2602, and 1032 activity instances from the first, second and the third apartments were represented by as few as 11, 10 and 9 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 didn't choose any specific sensor for that activity, and as a result the activity was eliminated. The same applied for "taking medicines" in third apartment. This shows how our algorithm can avoid mapping very irregular activities. It also shows how the algorithm condensed the activity instances into a compressed representation, as we approximately obtained the 11 predefined activities. During activity extraction, also the number of sensors for each activity template was reduced from an average of 32.13 sensors to 1.94 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.

Figure 3 shows the number of discovered activity templates for the target data. For example, using three days of unlabeled target data and no labeled target data, we discovered 8, 7, and 7 activity templates for the first, second and third apartments, respectively. The similarity threshold ς in those experiments was set to the midpoint 0.5. The reason that fewer activity templates are discovered is because some similar activities might be merged into one activity, such as relaxing and eating which happen at similar times and similar places. Also variations of the same activity that initially might have been considered as different activities, might be merged together as more data is provided. Also for some other activities it is not very easy to discover them from



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

(d) Apartment 1. (e) Apartment 2. (f) Apartment 3.

Figure 2: Figures (a-c) show sensor map and location tags for each apartment. Circle and triangles on the map show motion sensor and contact sensors. Figures (d-f) show distribution of residents' activities per each day (horizontal axis) for 3 months (vertical axis).

only a few days of data, such as housekeeping 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 be discovered.

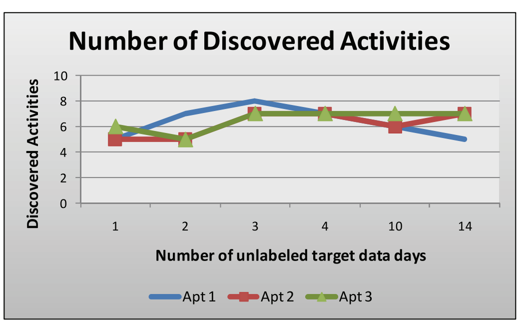


Figure 3: Number of discovered activities in the target space.

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's only used at the end to verify the correctness of the results. The mapping accuracy is defined as number of target activities which their assigned label matches the actual embedded label. Figure.4 shows the mapping accuracy for different amounts of unlabeled target data and no labeled target data, in several different settings. It's interesting to note that transferring activities from a bigger apartment such as the third apartment to smaller apartments such as the first apartment leads to better results (e.g. 83% vs. 67% for 3 days of unlabeled data). One explanation can be the lack of certain spaces in smaller apartments, such as the workspace in the first and second apartments. It should be noted that some activities might not be present in both spaces, such as working or housekeeping. Also transfer between the first and second apartment produced relatively satisfactory results, as those two apartments have a more similar layout and functional structure.

We tested two of our own activity recognition algorithms on the transferred labeled data. The first algorithm is a nearest neighborhood (NN) algorithm based on the similarity measure in Equation 2. The second algorithm is a standard hidden Markov model (HMM) which learns the activities using the Viterbi algorithm (Viterbi 1967). The models almost performed the same with the nearest neighborhood algorithm sometimes slightly outperforming HMM due to its use of temporal and spatial features. Using the embedded labels we define the recognition rate as the percentage of sensor events predicted with the correct label. Figure. 5 shows NN's recognition rate based on mapping from apartment 3 to 1 using 0 and 1 day of labeled target data. Figure. 6 shows

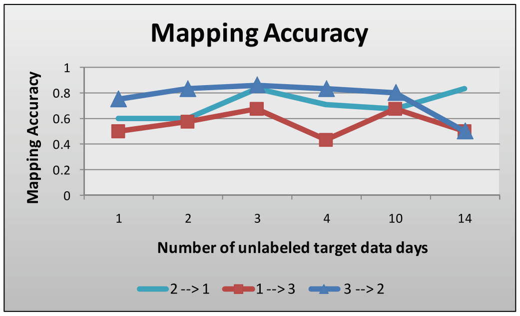


Figure 4: Mapping accuracy in several different settings.

recognition rate based on mapping from apartment 2 to 1 for both NN and HMM. Our results show that despite using little to no labeled target data, and having different layouts and schedules, both algorithms still perform recognition in a target space using data from a source space.

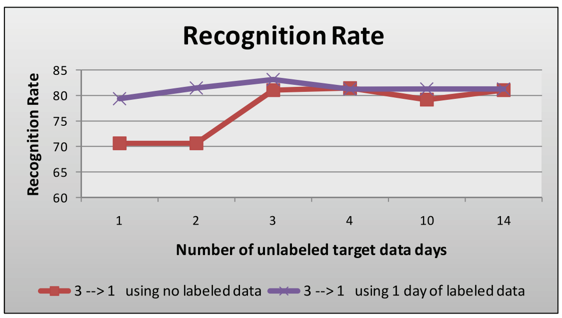


Figure 5: Nearest neighborhood's recognition rate based on mapping from apartment 3 to 1 using 0 and 1 day of labeled target data.

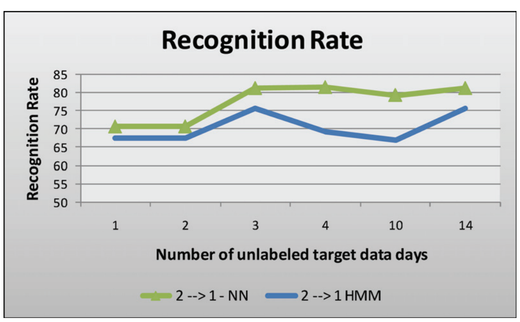


Figure 6: Recognition rate based on mapping from apartment 2 to 1 for nearest neighborhood and HMM.

Conclusion and Future Work

Activity recognition plays an important role in smart environments and is an essential component for planning and plan recognition in smart environments. This paper introduces a method of transferring learned activities from one physical space to another, in order to avoid the time consuming task of data annotation for each new physical space and to achieve a more accelerated deployment process. Our experiment results show that it's possible to recognize activities using no labeled data from the target space, and despite the fact that the apartment layouts and residents schedules were different.

In the future, we intend to combine this method with adaptive and active learning methods in order to be able to enhance the results over time. We also want to develop algorithms that can map activities from environments with totally different functionalities, such as from a workplace to a residential space. Ultimately, we intend to integrate our activity recognition system with a plan recognition and planning component as part of an assisted living project. Such an integrated system will allow us to effectively recognize elderly's activities and plans and provide timely prompts and cues, whenever necessary.

Acknowledgement

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

References

Abowd, G., and Mynatt, E. 2004. *Smart Environments: Technology, Protocols, and Applications*. Wiley. chapter Designing for the human experience in smart environments., 153–174.

Asadi, M., and Huber, M. 2007. Effective control knowledge transfer through learning skill and representation hierarchies. In *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*, 2054–2059.

Brdiczka, O.; Maisonnasse, J.; and Reignier, P. 2005. Automatic detection of interaction groups. In *Proceedings of the 7th international conference on Multimodal interfaces*, 32–36.

Can, F. 1993. Incremental clustering for dynamic information processing. *ACM Trans. Inf. Syst.* 11(2):143–164.

Caruana, R. 1997. Multitask learning. *Mach. Learn.* 28(1):41–75.

Cook, D., and Das, S. 2004. *Smart Environments: Technology, Protocols and Applications (Wiley Series on Parallel and Distributed Computing)*. Wiley-Interscience.

Cook, D.; Youngblood, M.; Heierman, E.O., I.; Gopalratnam, K.; Rao, S.; Litvin, A.; and Khawaja, F. 2003. Mavhome: an agent-based smart home. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, 521–524.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B* 39(1):1–38.

Dieter, H. K.; Fox, D.; Etzioni, O.; Borriello, G.; and Arnstein, L. 2002. An overview of the assisted cognition project. In *In AAAI-2002 Workshop on Automation as Caregiver: The Role of Intelligent Technology in Elder*.

Doctor, F.; Hagras, H.; and Callaghan, V. 2005. A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 35(1):55–65.

Gu, T.; Wu, Z.; Tao, X.; Pung, H.; ; and Lu, J. 2009. 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*.

Guyon, I., and Elisseeff, A. 2003. An introduction to variable and feature selection. *J. Mach. Learn. Res.* 3:1157–1182.

Helal, S.; Mann, W.; El-Zabadani, H.; King, J.; Kaddoura, Y.; and Jansen, E. 2005. The gator tech smart house: A programmable pervasive space. *Computer* 38(3):50–60.

Inomata, T.; Naya, F.; Kuwahara, N.; Hattori, F.; and Kogure, K. 2009. 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*, 39–42.

Levinson, R. 1997. Peat: the planning and execution assistant and trainer. *Journal of Head Trauma Rehabilitation* 769.

Liao, L.; Fox, D.; and Kautz, H. 2005. Location-based activity recognition using relational markov networks. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 773–778.

Maurer, U.; Smailagic, A.; Siewiorek, D. P.; and Deisher, M. 2006. 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*, 113–116. Washington, DC, USA: IEEE Computer Society.

Pan, S. J., and Yang, Q. 2008. A survey on transfer learning. Technical Report HKUST-CS08-08, Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China.

Pei, J.; Han, J.; and Wang, W. 2007. Constraint-based sequential pattern mining: the pattern-growth methods. *Journal of Intelligent Information Systems* 28(2):133–160.

Philipose, M.; Fishkin, K.; Perkowitz, M.; Patterson, D.; Fox, D.; Kautz, H.; and Hahnel, D. 2004. Inferring activities from interactions with objects. *Pervasive Computing, IEEE* 3(4):50–57.

Raina, R.; Ng, A. Y.; and Koller, D. 2006. Constructing informative priors using transfer learning. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, 713–720. New York, NY, USA: ACM.

Rashidi, P., and Cook, D. J. 2008. An adaptive sensor mining framework for pervasive computing applications. In *International Workshop on Knowledge Discovery from Sensor Data(Sensor-KDD 2008)*, 41–49.

Rashidi, P., and Cook, D. J. 2009a. the resident in the loop: Adapting the smart home to the user. *IEEE Transactions on Systems, Man, and Cybernetics journal, Part A* 39(5):949–959.

Rashidi, P., and Cook, D. J. 2009b. Transferring learned activities in smart environments. In *5th International Conference on Intelligent Environments*, volume 2 of *Ambient Intelligence and Smart Environments*, 185–192.

Rialle, V.; Ollivet, C.; Guigui, C.; and Hervé, C. 2008. 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* 47:63–9.

Roy, D. M., and Kaelbling, L. P. 2007. Efficient bayesian task-level transfer learning. In *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*, 2599–2604.

Simpson, R.; Schreckenghost, D.; LoPresti, E. F.; and Kirsch, N. 2006. *Designing Smart Homes*, volume 4008 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg. chapter Plans and Planning in Smart Homes, 71–84.

Sixsmith, A.; Meuller, S.; Lull, F.; Klein, M.; Bierhoff, I.; Delaney, S.; and Savage, R. 2009. Soprano: An ambient assisted living system for supporting older people at home. In *ICOST '09: Proceedings of the 7th International Conference on Smart Homes and Health Telematics*, 233–236. Berlin, Heidelberg: Springer-Verlag.

Tan, P.-N.; Steinbach, M.; and Kumar, V. 2005. *Introduction to Data Mining*. Adison Wesley.

Taylor, M. E.; Stone, P.; and Liu, Y. 2007. Transfer learning via inter-task mappings for temporal difference learning. *Journal of Machine Learning Research* 8:2125–2167.

Thrun, S. 1996. Is learning the nth thing any easier than learning the first? *In Advances in Neural Information Processing Systems* 8:640–646.

van Kasteren, T.; Englebienne, G.; and Krose, B. 2008. Recognizing activities in multiple contexts using transfer learning. In *AAAI AI in Eldercare Symposium*.

Viterbi, A. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on* 13(2):260 – 269.

Wren, C., and Munguia-Tapia, E. 2006. Toward scalable activity recognition for sensor networks. In *Proceedings of the Workshop on Location and Context-Awareness*, 218–235.

Zheng, V. W.; Hu, D. H.; and Yang, Q. 2009. Cross-domain activity recognition. In *Ubicomp '09: Proceedings of the 11th international conference on Ubiquitous computing*, 61–70. New York, NY, USA: ACM.