Automated activity-aware prompting for activity initiation

Lawrence B. Holder PhD[a]
E: holder@eecs.wsu.edu

Diane J. Cook PhD[a]
E: cook@eecs.wsu.edu

[a]Washington State University, Pullman, WA, USA

L.B. Holder, D.J. Cook. Automated activity-aware prompting for activity initiation. Gerontechnology 2013;11(4):534-544; doi:10.4017/gt.2013.11.4.005.00 Performing daily activities without assistance is important to maintaining an independent functional lifestyle. As a result, automated activity prompting systems can potentially extend the period of time that adults can age in place. In this paper we introduce AP, an algorithm to automate activity prompting based on smart home technology. AP learns prompt rules based on the time when activities are typically performed as well as the relationship between activities that normally occur in a sequence. We evaluate the AP algorithm based on smart home datasets and demonstrate its ability to operate within a physical smart environment.

Keywords: activity prompting, pervasive computing, activity recognition, smart home

There will be dramatic growth in the aging population over the next 40 years[1] as well as shortages in healthcare resources and personnel[2]. Given the prohibitive costs of formal healthcare and institutionalization, along with older adults' desire to 'age in place'[3], there is a mounting need for the development of assistive technologies to extend the amount of time individuals can live independently in their homes. In recent years, rapid advancements have been made in the development of assistive smart environment technologies geared toward increasing older adults' functional independence and improving health outcomes and well-being. These technologies include socially and physically assistive robots[4,5], unobtrusive in-home monitoring[6], complex activity recognition[7], home telecare[8], and reminder systems[9].

As the general population ages, the number of older adults with mild cognitive impairment (MCI) is growing[10]. MCI has been defined as an intermediate state between normal aging and dementia[11] and is characterized by impairments greater than expected for age in memory and other cognitive abilities with relative sparing of functional abilities. Despite intact abilities to carry out basic functional tasks, people with MCI often experience difficulty carrying out instrumental activities of daily living (IADLs), which are cognitively complex functional tasks like using the telephone, preparing meals, taking medications, and managing money[12]. Activities dependent on memory, executive functioning, and working memory such as medicine use and financial management tend to be most difficult for individuals with MCI[13,14]; however, in order to function independently at home, individuals need to be able to complete these IADLs[15].

When individuals with cognitive impairment fail to initiate or complete everyday IADLs, caregivers are often responsible for monitoring IADLs and providing reminders or prompts as needed. Broadly defined, 'prompts' are any form of verbal or non-verbal intervention delivered to an individual on the basis of time, context, or acquired intelligence that helps in successful completion of an activity[9]. These are time consuming and burdensome tasks that are often associated with negative effects for the caregiver's own health[16]. Smart environment technologies that help people with MCI carry out their IADLs by detecting when assistance is needed and automatically delivering reminders or prompts have the potential to reduce caregiver burden and allow aging adults to retain their functional independence longer.

A smart environment is any physical environment (for instance, home, workplace, shopping mall, hospital) that senses the state of the resident and the physical surroundings and acts in order to ensure the well-being of the resident and the environment[17]. Research in smart environments has gained popularity in the last decade and the potential use of smart home technology for health monitoring in an individual's own home is viewed as 'extraordinary'[18]. The goal of the

Figure 1. CASAS 'smart home in a box' kit (left) and smart home installation site (right)

CASAS (Center for Advanced Studies in Adaptive Systems) smart home project at Washington State University is to design a 'smart home in a box' that is simple to install and performs key functions such as activity recognition, monitoring, and prompting that customize behavior to the resident with little or no effort on the part of the resident.

In this paper, we describe a new method to perform smart home-based automated activity prompting that customizes its behavior to the resident with no input on their part. Specifically, our technology utilizes data collected from an individual's home to learn rules for prompting the individual to initiate important daily activities such as taking medicine, exercising, calling their children, or any other activity for which data is available. Reminder systems have long been in existence and range from simple alarm clocks to complex systems that are based on rules, planning or machine learning. Rule-based reminder systems allow a user to specify rules based on time, context and preferences[19,20]. More adaptive reminder systems integrate reinforcement learning[21], which requires a pre-specified complete schedule of activities but can make adjustments without direct user feedback. Other approaches use dynamic Bayesian networks[22], Markov decision processes[23], and Markov-based planning[24] to coordinate and give time prompts for these pre-scheduled activities. Active learning has been employed as well[25] to interactively manage calendar synchronization.

While reminder systems have been widely explored, few take into account an individual's behavioral patterns to provide context-aware prompts, despite the fact that studies indicate activity-aware prompts offer significant advantages over traditional time-based prompts[26]. Our unique contribution to the area of prompting systems is to design an approach that is completely automated, based on activity recognition. We assume that sensor data is collected in a home while an individual performs his or her daily routine. We also assume there are instances of the activities requiring prompting (when the individual correctly performed the activity or were prompted by a caregiver to initiate the activity). Our algorithm, called AP for Activity Prompting, learns rules that define when the activities normally occur and utilizes these rules to automate prompting. We evaluate our algorithm based on real data collected in CASAS smart environments.

Methods

Our AP activity prompting system is designed as part of a larger CASAS smart home project. We define a smart environment as an intelligent agent that utilizes information collected about the resident and the physical surroundings to improve the experience of the individual in the environment[17]. We design the CASAS smart home to be easily installed and usable without customization. The CASAS 'smart home in a box' kit fits within a single small box (*Figure 1*). The box contains wireless infrared motion sensors and magnetic door sensors that are placed throughout the home. The sensors generate event messages when motion or door usage is detected. Messages are collected via the CASAS mesh network, processed by the CASAS publish/subscribe middleware, and stored in an SQL (Structured Query Language) database on a small, low-power computer. To date, we have installed over 30 smart home kits. Installation takes approximately two hours and removal is completed in 30 minutes.

In order to learn activity prompt timings directly from sensor data, the AP activity prompting algorithm operates together with another software agent, called AR (Activity Recognition). Sensor events occurring in the environment are passed to AR, which assigns an activity label to the event.

These activity labels are passed to AP, which provides the context necessary to decide if a prompt is necessary. Both systems are trained on labeled sensor data. AR learns classifiers for predicting the activity label of a sensor event. AP learns patterns for predicting when an activity will occur relative to other activities and time landmarks. Here we describe the AR and AP algorithms.

Activity Recognition (AR)

Any smart environment that focuses on the needs of its residents requires information about the activities that are being performed by the resident. At the core of these systems, then, is activity recognition, which is a challenging and well-researched problem[27,28]. Activity recognition plays a critical role with prompting. First, some activities requiring prompting are correlated with other activities (for instance, washing dishes occurs after eating, medicine should be taken during dinner). Secondly, an intelligent prompter needs to recognize when the prompted activity has been performed and suppress further prompting in these situations.

The goal of activity recognition is to map a sequence of sensor data to a corresponding activity label. The CASAS activity recognition software, called AR, provides real-time activity labeling as sensor events arrive in a stream. To do this, we formulate the learning problem as that of mapping the sequence of the *k* most recent sensor events to a label that indicates the activity corresponding to the last (most recent) event in the sequence. The sensor events preceding the last event define the context for this last event.

Data collected in a smart home consists of events generated by the sensors. These are stored as a 4-tuple: <Date, Time, SensorId, Message>. For example, consider the following sequence of sensor events.

2011-06-15 03:38:23.271939 BedMotionSensor ON
2011-06-15 03:38:28.212060 BedMotionSensor ON
2011-06-15 03:38:29.213955 BedMotionSensor ON

These events could be mapped to a 'Sleep activity' label. To provide input to the classifiers, we define features describing data point i that correspond to a sensor event sequence of length k. The vector x_i includes values for 25 features (*Table 1*). Each label y_i corresponds to the activity label associated with the last sensor event in the window. A collection of x_i and the corresponding y_i are fed into a classifier to learn the activity models in a discriminative manner, i.e., a classifier is learned that map a sensor event sequence to a corresponding activity label. Although a fixed window size k could be identified that works well for a given data set, AR dynamically adjusts the window size based on the most likely activities that are being observed and the activity duration that is typical for those activities.

Our AR algorithm uses a support vector machine (SVM) method for real-time activity recognition[29]. A support vector machine identifies a hyperplane (or set of hyperplanes) which separates points into different classes with the largest possible distances between the hyperplanes and the data points. Researchers have reported results from alternative machine learning models[30-34], including Bayes classifiers, hidden Markov models, decision trees, and conditional random fields. In an earlier experiment[35] we tested multiple models for their ability to recognize activities in real time from streaming data. We found that SVMs consistently achieved the strongest performance in those cases and thus use SVMs as the basis for the activity recognition approach described in this paper. In addition, SVMs offer advantages in terms of determining the degree of fit between a data point and a class, which can be useful when identifying anomalous points. However, SVMs are costly in terms of training time and thus methods are needed to reduce these costs.

We use the LibSVM implementation[36] with the one-vs-one paradigm and a radial basis function kernel with default parameter settings. The one-vs-one paradigm learns a set of classifiers that distinguish every pair of classes. The classifier with the highest output function assigns the identified class label to the data point. Experiments conducted by Hsu and Lin[37] reveal that the one-vs-one paradigm is one of the most practical multi-class SVM approaches. Though we test our methodology in this paper using an SVM, our methodology can make use of any classifier.

Representing prompt rules

The goal of our automated prompting system is to automatically learn rules that describe when an activity is typically initiated. Once the rules are learned, they can be used to issue prompts at the appropriate time or context. Activities that are part of an individual's regular routine are usually initiated based on wall-clock time or based

Table 1. The feature vector describing a data point

Feature #	Value
1..16	#Times each sensor generated an event in the sequence (16 unique sensors)
17..20	Time of day at the beginning of the sequence (morning, afternoon, evening, night)
21..24	Time of day at the end of the sequence
25	Time duration of the entire sequence

on activity context. As an example of the first pattern, one of our smart home residents requested reminders to 'pick up her grandchildren from school at 3pm every Tuesday afternoon'. However, she also needed a reminder to 'take her medicine while eating breakfast', which is an example of the second type of prompting pattern.

AP learns patterns for each prompting activity, *PA,* as a function of another reference activity, *RA,* with which it is highly time-correlated. AP models the relative time offset between initiation of *RA* and *PA* as a Gaussian distribution with a corresponding mean and standard deviation. The specific format for an activity pattern is thus:

<activity> [<relative_activity> <mean (s)> <standard_deviation (s)>]+

where the "+" means one or more relative activities.

Consistent with the earlier discussion, the possible relative activities for *PA* include all other activities the resident performs, combined with periodic clock-based activities. The periodic clock-based activities include the start of each year, month, day, week, and hour, as well as the start of each specific month of the year (January, February, .., December) and each specific day of the week (Sunday, .., Saturday). This way, activity timings can be learned both for activities that occur at regular times and activities whose occurrence is relative to another activity. Each activity pattern is represented by the name of the prompted activity *PA*, the name of the relative activity *RA*, the mean time delay in seconds between *RA* and *PA*, and the time standard deviation in seconds. For example, if the activity 'Pick up grandchildren' takes place every Tuesday around 2:40pm (+/- 5 minutes), the associated prompt pattern would be represented as:

Pick_up_grandchildren Activity_Tuesday 52800 300

If the individual picked up her grandchildren every Tuesday and Thursday around 2:40pm, then the pattern would be:

Pick_up_grandchildren Activity_Tuesday 52800 300 Activity_Thursday 52800 300

On the other hand, if the individual needs a reminder to take medicine about ten minutes (+/- 5 minutes) after breakfast begins each morning, then the corresponding pattern would be:

Take_Medicine Eat_Breakfast 600 300

Learning prompt rules

For each prompting activity *PA*, AP learns a prompt rule using a two-step process: consider patterns based on a single relative activity, and then consider patterns based on multiple relative activities. All of these possibilities are evaluated (as described below) and the highest-ranked pattern is chosen for the prompting rule. First, we consider the method for evaluating patterns based on a single relative activity. AP must select a relative activity other than *PA* from among the activities the resident performs, along with the periodic clock-based activities described earlier. An ideal relative activity *RA* is one that always occurs before each instance of the prompting activity *PA* and always at the same (ideally small) time before *PA*. Therefore, the score for a relative activity *RA* should increase proportional to the number of times it co-occurs with *PA*, should decrease proportional to the variance in the time delay between each *RA* and *PA*, and should decrease proportional to the absolute time delay between each *RA* and *PA*. Therefore, each potential relative activity *RA* is evaluated according to three properties: (i) the likelihood that activity *PA* occurs after each activity *RA*, (ii) the confidence in the distribution of the occurrence times of *PA* relative to *RA*, and (iii) the mean delay between *RA* and *PA*.

Combining all these factors, we arrive at the following promptability measure *P*:

$$P = \left(\frac{n}{m}\right)\left(\frac{1}{\sigma/\sqrt{n}}\right)\left(\frac{1}{\sqrt{m-n}}\right)\left(\frac{1}{\sqrt{\mu}}\right) \qquad [1]$$

Property 1 is essentially the probability that *RA* occurs before each instance of *PA*. We estimate this probability from the dataset. Given *m* instances of relative activity *RA* in the sensor data, and *n* instances of activity *PA* occurring between two consecutive *RA*s, we estimate the occurrence likelihood as *n/m*. This forms the first factor of our overall promptability score *P* [1] for *RA* as the relative activity for *PA*.

Property 2 measures the variance in the delay between the two activities. Again, we want minimal variance, so this factor will be in the denominator of [1]. There are two contributions to the variance in the delays. The first contribution is the actual variance in the distribution of the delays between each co-occurrence of *RA* and *PA*. Over all such occurrences of *PA* preceded by *RA*, AP models the time delay (in seconds) between the two activities as a Gaussian and computes the corresponding mean μ and standard deviation σ for these delays. We use the standard error as an estimate of the confidence (smaller the better) that *PA* follows μ seconds after *RA*. This comprises the second factor in P below, which decreases P based on increased distribution error. The second contribution to the delay error involves the $(m\text{-}n)$ occurrences of RA that are not followed by an occurrence of PA. We estimate this contribution to the distribution error as the standard error based on a variance of one and a sample size of $(m\text{-}n)$. This comprises the third factor in P below, which decreases P based on increased distribution error due to the absence of a PA after RA.

Property 3 prefers a smaller mean delay time μ. Therefore, we include the fourth factor in P below, which decreases P as the mean delay increases.

The resulting promptability measure (P) estimates the correlation between the two activities and thus represents how well RA plays the role of the relative activity for PA. If $m=0$ or $n=0$, we set $P=0$. If $\sigma=0$, we set $\sigma=1$. If $\mu=0$, we set $\mu=1$. If $(m\text{-}n)=0$, we set $(m\text{-}n)=1$. The relative activity with the highest P value, along with its associated mean and standard deviation, are output as the activity's pattern. If two relative activities have the same P value, we prefer the one with the smaller mean.

The second step in the process is to consider rules where the prompt activity PA occurs relative to several other relative activities, not just one. While prompt patterns can be learned as a function of multiple relative activities, considering all subsets of activities as potential patterns is not computationally tractable. However, AP does consider such patterns involving subsets of the months of the year (January, February, …, December), the days of the week (Sunday, Monday, …, Saturday), and the hours of the day, since many activities are scheduled relative to specific sets of months, days or hours (for instance, leaving for work at 7am Monday through Friday). To accomplish this, we consider three additional relative activities: month-of-year, day-of-week and hour-of-day, where their promptability P values are computed as the sum of the above-average P values of each individual month, day or hour within the set. If one of these multiple relative activity patterns wins out over all the others, then the output pattern consists of all the individual month, day or hour relative activities whose frequency is in the upper half of the range of normalized frequencies. So, using our example of leaving for work at 7am Monday through Friday, we would consider the day-of-week relative activity by summing the above-average P values for each individual day of the week. AP would detect that the frequencies for Sunday and Saturday are low, and these days are thus not included. Therefore, the final pattern would look as follows (assuming 7am +/- 15 minutes):

Leave_for_Work	Activity_Monday 25200 900	Activity_Tuesday 25200 900
	Activity_Wednesday 25200 900	Activity_Thursday 25200 900
	Activity_Friday 25200 900	

Monitoring and prompting
In monitoring mode, AP determines if a prompt should be issued for one of the activities based

Figure 2. CASAS prompts can be sent to a touch-screen device (left) or to a mobile device (right). The text and audio content can be automatically generated, and the user may choose to respond to the prompt or ignore it

on the current time, the activity context, and the prompting rules that were learned. When a prompt is needed then AP sends a message to the CASAS middleware with a prompt command and (optionally) the prompt message. The CASAS middleware then handles issuing the prompt to the appropriate device, such as a touch-screen computer located in the home or a mobile device (*Figure 2*).

In order to determine if a prompt should be issued, AP needs two pieces of information: the current date and time and occurrences of activities as detected by AR. The CASAS middleware issues a chime each minute which updates AP on the current date and time and can use this to monitor appropriate time-based relative activities. The CASAS middleware also forwards AR-generated activity labels that AP uses to determine if a prompted activity *PA* has been performed or if its relative activity *RA* has been initiated.

If the prompter detects the occurrence of a relative activity *RA* that is referenced in the pattern of one of the prompt activities *PA*, the prompter then watches for the beginning of *PA*. If *PA* does not occur within $(\mu-\sigma)$ seconds, where μ and σ are the mean and standard deviation time delay based on the pattern, then the prompter issues a first prompt for the user to execute this activity *PA*. If another σ seconds go by without activity *PA* being observed by AR, then the prompter issues a second prompt for *PA*. Finally, if yet another σ seconds go by without observing activity *PA*, the prompter issues a third and final prompt for *PA*, and then returns to watching for the next occurrence of the relative activity.

The prompts are sent to the CASAS middleware and consist of the activity name and the prompt repetition level (1, 2 or 3). The system assumes that these prompt messages will be appropriately handled by another process, which will most likely play the audio file named <activity>_<level>.mp3, and then provide a pop-up alert to the user with pre-specified response buttons (for instance, 'OK', 'Later', 'No'). The text, image and audio files can be provided in advance by the user or the user can decide to use messages that are automatically generated by AP. Finally, AP will cancel prompts if AR detects the activity has been initiated or the user selects an appropriate response.

Results

We validate the ideas described in this paper on sensor event datasets collected in CASAS smart homes. First, we want to evaluate the ability of AP to learn prompting rules that reflect accurate times when the activity would be expected to occur. Second, we want to evaluate the ease with which AP operates in the CASAS environment. To address the first goal we test the AP algorithm on sensor event datasets collected from three smart apartment testbeds (*Figure 3*). Each of the smart apartments housed one older adult (age 65+) resident with no evidence of cognitive impairment. During the six months that we collected data in each of the apartments, the residents lived in the apartments and performed their normal daily routines.

Figure 3. The sensor layouts (black circles) and doors (black rectangles) for the three apartments: B1 (top left), B2 (top right), and B3 (bottom)

Human annotators tagged sensor events with the corresponding activities to provide ground truth for our evaluation, after looking at a visualization of the data and interviewing residents to gain insights on their daily activities. The 12 activities that were annotated (*Table 2*), and an activity occurrence is defined as an uninterrupted sequence of sensor events annotated with that activity. The AR activity recognition algorithm was tested using three-fold cross validation for each of the datasets B1, B2 and B3, and reporting accuracy as a ratio between the number of data points correctly classified to the total number of data points.

To evaluate AP's prompting performance, for each dataset we used the first 2/3 of the data to learn the prompt

Table 2. Number of activity occurrences, total sensor events for each activity, and accuracy classification using the support-vector-machine based algorithm implementation for datasets B1, B2, and B3, taken over a 6-8 month time period

Activity	Data sets with collection period					
	B1: 7/17/09 – 2/3/10		**B2:** 6/15/09 – 2/4/10		**B3:** 8/11/09 – 2/4/10	
	Occurrences	**Events**	**Occurrences**	**Events**	**Occurrences**	**Events**
Bathing	84	7,198	74	16,295	48	5,151
Bed to toilet	136	4,170	353	14,641	119	4,346
Cook	874	101,820	593	55,240	580	44,842
Eat	556	28,771	415	24,417	418	39,453
Enter home	584	3,711	462	2,440	179	996
Housekeeping	65	3,280	255	12,971	0	0
Leave home	577	4,305	460	2,476	211	1,246
Personal hygiene	1,042	39,190	938	42,704	605	37,237
Relax	642	39,934	199	16,996	107	8,207
Sleep	336	33,213	406	10,477	299	20,693
Take medicine	587	15,388	170	22,524	64	1,248
Work	0	0	0	0	393	108,763
Classification accuracy	0.85		0.83		0.88	

rules and tested the prompt timings on the remaining data. For each dataset we report performance in terms of True Positives (the individual performed the prompted activity at the time predicted by the prompting rule), False Positives (the individual did not perform the prompted activity at the predicted time), and False Negatives (the individual performed the prompted activity at a time not predicted by the prompting rule). We do not report True Negatives, because they are difficult to define in this context.

In order to generate a baseline for comparison, we implemented two alternative mechanisms for generating prompts. The first is a time-based prompt (TB), which is a prompting strategy that has been considered by other researchers[27]. Using the time-based strategy, the start time for each prompting activity *PA* is estimated based on a Gaussian distribution over the number of seconds the activity is initiated past midnight. The second, activity-based approach (AB) utilizes only activity sequence information, similar to an approach that was introduced by researchers to predict UNIX commands[38]. In this approach, a relative activity *RA* is identified for each prompting activity *PA*, where *RA* is the activity that occurs most often just prior to *PA*. A Gaussian distribution is then used to model the relative time between the initiations of the two activities. In both cases, initial prompts would be delivered at the distribution mean minus one standard deviation, then at the distribution mean, and then at the distribution mean plus one standard deviation.

In all three datasets, the AP approach results in more true positives and fewer false negatives than either the TB or AB approaches (*Table 3*). The TB approach had fewer false positives than AP or AB, mainly because TB can only predict activities once per day. So, while TB will perform poorly for activities that occur more than once per day, TB will also avoid more false positives when activities occur only once per day or less often. AP has fewer false positives than AB in datasets B1 and B2, but more false positives in dataset B3. Overall, most actual occurrences of an activity are correctly predicted by AP, but the patterns learned by AP tend to over-generalize (i.e., more false positives). The activity occurrences that AP did not predict would occur seem to be mostly due to the high variance in the start time of most activities.

In our final experiment we integrate AP into the CASAS middleware and test the prompt learning and generation ability in a physical smart environment setting. In this case, we introduced two activities that are performed on a regular basis: 'LeaveLab' and 'PrepareForMeeting'. In this setting AP received sensor messages from the CASAS middleware as well as chimes for each minute that passed. When the time arrived to prompt for one of the selected activities, AP sent a prompt message with the corresponding activity name to the CASAS middleware. CASAS, in turn, sent the corresponding prompt text and audio files to the touch screen device (*Figure 2*). The users in the lab were able to select a response on the touch screen if desired. AP correctly repeated the prompts until the activity was detected through AR, all of the prompts had been issued, or it heard a message to cancel the prompt through the middleware because of

Table 3. Results of three approaches for the three types of activity patterns (TP, FP, FN) on three dataset (B1-3); TP=True-positives; FP=False-positives; FN=False-negatives

Activity	Count	Algorithm AP			Time-based			Activity-based		
		TP	FP	FN	TP	FP	FN	TP	FP	FN
Dataset B1										
Bathing	16	16	127	0	16	63	0	16	384	0
Bed to toilet	31	31	149	0	31	48	0	31	77	0
Cook	301	249	1	52	79	0	222	139	14	162
Eat	231	194	6	37	79	0	152	215	86	16
Enter home	151	136	52	15	70	9	81	140	2	11
Housekeeping	41	1	0	40	26	53	15	30	123	11
Leave home	142	121	45	21	70	9	72	131	270	11
Personal hygiene	401	159	1	242	79	0	322	107	17	294
Relax	125	93	113	32	56	23	69	90	211	35
Sleep	109	70	9	39	77	2	32	31	0	78
Take medicine	154	124	23	30	76	3	78	140	161	14
Work	0	0	0	0	0	0	0	0	0	0
TOTAL dataset B1	1702	1194	526	508	659	210	1043	1070	1345	632
Dataset B2										
Bathing	28	28	170	0	24	44	4	27	319	1
Bed to toilet	110	110	45	0	64	4	46	106	48	4
Cook	234	136	6	98	67	1	167	170	176	64
Eat	166	160	9	6	68	0	98	160	74	6
Enter home	167	142	9	25	66	2	101	164	1	3
Housekeeping	43	42	218	1	24	44	19	42	123	1
Leave home	166	147	11	19	67	1	99	141	206	25
Personal hygiene	347	225	2	122	68	0	279	141	26	206
Relax	43	34	143	9	28	40	15	35	311	8
Sleep	155	147	7	8	67	1	88	106	4	49
Take medicine	70	67	20	3	67	1	3	67	166	3
Work	0	0	0	0	0	0	0	0	0	0
TOTAL dataset B2	1529	1238	640	291	610	138	919	1159	1454	370
Dataset B3										
Bathing	15	13	171	2	13	57	2	15	207	0
Bed to toilet	13	13	408	0	11	59	2	12	53	1
Cook	189	149	17	40	69	1	120	133	9	56
Eat	143	105	31	38	64	6	79	133	56	10
Enter home	46	37	33	9	32	38	14	45	2	1
Housekeeping	0	0	0	0	0	0	0	0	0	0
Leave home	47	36	34	11	32	38	15	42	179	5
Personal hygiene	222	141	4	81	69	1	153	55	11	167
Relax	68	55	107	13	40	30	28	20	26	48
Sleep	66	52	54	14	42	28	24	12	1	54
Take medicine	29	28	145	1	20	50	9	25	196	4
Work	132	114	63	18	60	10	72	105	84	27
TOTAL dataset B3	970	743	1067	227	452	318	518	597	824	373

a user response. This experiment demonstrates that the AP system can successfully work with the smart home components to interact with residents in delivering a customized prompt. Although the prompt was delivered to a touch screen in this experiment, it can be delivered to other media as well, including mobile devices. We will evaluate the effectiveness of the technology for prompting in a mobile setting as part of our future work.

Discussion

The results of our experiments indicate that activities can be predicted when data is available that is consistent with the normal desired behavioral routine. The need for activity prompting is particularly great for individuals with memory impairment. However, such individuals are less likely to have a a large corpus of complete, normal routine data from which activity timings can be learned. For practical use, a data-driven activity prediction system would need to be implemented before memory difficulties arise in order to obtain the necessary data. Alternatively, a caregiver could prompt the individual through a desired routine on a daily basis until sufficient complete data is collected. The amount of data that is required to identify activities and learn their timings depends on the number of activities being monitored and the normal variation in an individual's routine. Anecdotally, we have found that two weeks of complete data is sufficient for modeling the set of activities analyzed in this paper.

This work represents one of the first attempts to perform data-driven automation of activity prompting. There are a number of methods that can be explored to refine the approach that is described here. We note, for example, that the number of false positives generated by AP is high, which might be regulated by only issuing prompts with a high confidence value based on activity occurrence frequency. Furthermore, the learned rules are based on two main factors: selection of a relative activity and modeling of time offsets between the predicted activity and the relative activity. In many situations there could be a number of additional factors including time offset from secondary relative activities, the state of the environment, and external events such as weather and holidays. We will explore alternative learning and forecasting algorithms that can identify likely activity occurrence times from this larger set of influencing features.

Another factor that influences performance is the robustness of activity recognition. In this paper, we evaluated AR and AP in single-resident homes. The task of activity recognition becomes more challenging in homes with multiple residents and pets. The performance of an activity recognition algorithm will also depend upon the number, the complexity, and the similarity of activities that need to be distinguished and tracked. We will continue to refine and evaluate activity recognition in increasingly complex settings. We will also consider the usefulness of AR and AP using smart phone sensors instead of, or in addition to, smart home environmental sensors.

Predicting the timing of activities in order to deliver automated prompts is a relatively new area of investigation. In addition to designing techniques to address this problem, work is also needed to define appropriate performance measures. In this study we evaluated performance using historic data and based on whether the activity was performed at the predicted time or not. In the future we will consider measures that offer greater sensitivity such as the actual time difference between activity prompt and activity performance and the direction of the error (overly-anticipatory or overly-delayed prompts). We also intend to test AP in homes with older adult residents to evaluate the usefulness of the prompt timings and delivery mechanisms.

Conclusions

In this paper we introduce an algorithm to automate the creation of rules to prompt individuals for activity initiation. Such prompts can be valuable for individuals who have difficulty remembering important daily activities or who want to introduce new healthy behaviors into their routine. Unlike previous approaches, AP automatically creates prompt rules from data that identify prompt timings based on the wall-clock time the activity is typically performed together with its temporal relation to other activities. Because AP is part of a larger smart home project, it utilizes sensor events and activity recognition software to learn the rules, to issue the automated prompts, and to monitor whether the prompted activities are performed.

We evaluate the performance of AP on smart home datasets and find that our automated prompt timings outperform prompts that are based solely on wall-clock time or on activity sequences. We also demonstrate how AP can be integrated and used within the larger CASAS smart home system to automatically generate prompts in a physical smart environment setting.

Acknowledgements

The authors would like to thank Jim Kusznir, Brian Thomas, Maureen Schmitter-Edgecombe, and all the members of the CASAS team for their contributions to this work. This material is based upon work supported by the National Science Foundation under Grant Number 0852172, by the Life Sciences Discovery Fund, and by NIBIB Grant Number R01EB009675.

References

1. Vincent G, Velkoff V. The next four decades – the older population in the United States: 2010 to 2050. Washington: US Census Bureau; 2010; doi:10.1001/jama.2011.1221
2. Salzhauer A. Is there a patient in the house? Har-

vard Business Review; November 1, 2005
3. Eckert JK, Morgan LA, Swamy N. Preferences for receipt of care among community-dwelling adults. Journal of Aging and Social Policy 2004;16(2):49-65
4. Broekens J, Heerink J, Rosendal H. Assistive social robots in elderly care: A review. Gerontechnology 2009;8(2):94-103; doi:http://dx.doi.org/10.4017/gt.2009.08.02.002.00
5. Brose SW, Weber DJ, Salatin BA, Grindle GG, Wang H, Vazquez JJ, Cooper RA. The role of assistive robotics in the lives of persons with disability. American Journal of Physical Medicine and Rehabilitation 2010;89(6):509-521
6. Hayes TL, Pavel M, Kaye JA. An unobtrusive in-home monitoring system for detection of key motor changes preceding cognitive decline. In: Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society; 2004; pp 2480-2483; doi:10.1109/IEMBS.2004.1403715
7. Singla G, Cook DJ, Schmitter-Edgecombe M. Recognizing independent and joint activities among multiple residents in smart environments. Ambient Intelligence and Humanized Computing Journal 2010;1(1):57-63; doi:10.1007/s12652-009-0007-1
8. Botsis T, Demiris G, Pedersen S, Hartvigsen G. Home care technologies for the elderly. Journal of Telemedicine and Telecare 2008;14(7):333-337; doi:10.1258/jtt.2008.007002
9. Das B, Dook DJ, Schmitter-Edgecombe M, Seelye A. PUCK: An automated prompting system for smart environments. Personal and Ubiquitous Computing 2012;16(7):859-873
10. Dlugaj M, Weimar C, Wege N, Verde PE, Gerwig M, Dragano N, Moebus S, Jockwel KH, Erbel R, Siegrist J. Prevalence of mild cognitive impairment and its subtypes in the Heinz Nixdorf Recall study cohort. Dementia and Geriatric Cognitive Disorders 2010;30(4):362-373; doi:10.1159/000320988
11. Winblad B, Palmer K, Kivipelto M, Jelic V, Fratiglioni L, Wahlund LO, Nordberg A, Backman L, Albert M, Almkvist O, Araj H, Basun H, Blennow K, de Leon M, DeCarli C, Erkinjuntti T, Giacobini E, Graff C, Hardy, J, Jack C, Jorm A, Ritchie K, van Duijn C, Visser P, Petersen RC. Mild cognitive impairment – beyond controversies, toward a consensus: Report of the International Working group on Mild Cognitive Impairment. Journal of Internal Medicine 2004;256(3):240-246; doi:10.1111/j.1365-2796.2004.01380.x
12. Albert MS, DeKosky ST, Dickson D, Dubois B, Feldman HH, Fox NC, Garnst A, Holtzman DM, Jagust WJ, Petersen RC, Snyder PJ, Carrillo MC, Thies B, Phelps CH. The diagnosis of mild cognitive impairment due to Alzheimer's disease: Recommendations from the National Institute on Aging – Alzheimer's Association Workgrounds on diagnostic guidelines for Alzheimer's disease. Alzheimer's and Dementia: The Journal of the Alzheimer's Association 2011;7(3):270-279
13. Allaire JC, Gamaldo A, Ayotte BJ, Sims R, Whitfield K. Mild cognitive impairment and objective instrumental everyday functioning: The everyday cognition battery memory test. Journal of the American Geriatric Society 2009;57(1):12-125; doi:10.1111/j.1532-5415.2008.02054.x
14. Schmitter-Edgecombe M, Woo E, Greeley D. Characterizing multiple memory deficits and their relation to everyday functioning in individuals with mild cognitive impairment. Neuropsychology 2009;23(2):168-177; doi:10.1037/a0014186
15. Diehl M, Marsiske M, Horgas AL, Rosenberg A, Saczynski JS, Willis SL. The revised observed tasks of daily living: A performance-based assessment of everyday problem solving in older adults. Journal of Applied Gerontechnology 2005;24(3):211-230; doi:10.1177/ 0733464804273772
16. Vitaliano PP, Echeverria D, Yi J, Phillips PEM, Young H, Siegler IC. Psychophysiological mediators of caregiver stress and differential cognitive decline. Psychology and Aging 2005;20(3):402-411; doi:10.1037/0882-7974.20.3.402
17. Cook DJ. How smart is your home? Science 2012;335(6076):1579-1581; doi:10.1126/science.1217640
18. Department of Health. Speech by the Right Honourable Patricia Hewitt MP, Secretary of State for Health; http://collections.europarchive.org/tna/20051219093944/dh.gov.uk/en/news/speeches/speecheslist/dh_4124484; retrieved March 7, 2013
19. Lim M, Choi J, Kim D, and Park S. A smart medication prompting system and context reasoning in home environments. In: Proceedings of the Fourth IEEE International Conference on Networked Computing and Advance Information Management; 2008; pp 115-118; doi:10.1109/NCM.2008.254
20. Oriani M, Moniz-Cook E, Binetti G, Zanieri G, Frisoni G, Gerold C, De Vreese L, Zanetti O. An electronic memory aid to support prospective memory in patients in the early stages of Alzheimer's disease: A pilot study. Aging and Mental Health 2003;7(1):22-27; doi:10.1080/1360786021000045863
21. Rudary M, Singh S, Pollack M. Adaptive cognitive orthotics: Combining reinforcement learning and constraint-based temporal reasoning. In: Proceedings of the International Conference on Machine Learning; 2004; pp 91-98; doi:10.1145/1015330.1015411
22. Pollack M, Brown L, Colbry D, McCarthy C, Orosz C, Peintner B, Ramakrishnan S, Tsamardinos I. Autominder: An intelligent cognitive orthotic system for people with memory impairment. Robotics and Autonomous Systems 2003;44(3-4):273-282; doi:10.1.1.18.5664
23. Pineau J, Montemerlo M, Pollack M, Roy N, Thrun S. Towards robotic assistants in nursing homes: Challenges and results. Robotics and Autonomous Systems 2003;42(3-4):271-281; doi:10.1.1.12.6002
24. Boger J, Poupart P, Hoey J, Boutilier C, Fernie G, Mihailidis A. A decision-theoretic approach to task assistance for persons with dementia. In:

Proceedings of the International Joint Conference on Artificial Intelligence; 2005; p 1293; doi:10.1.1.72.9543
25. Weber J, Pollack M. Entropy-driven online active learning for interactive calendar management. In: Proceedings of the International Conference on Intelligent User Interfaces; 2007; p 141-10; doi:10.1.1.111.1319
26. Kaushik P, Intille S, Larson K. User-adaptive reminders for home-based medical tasks. A case study. Methods of Information in Medicine 2008;47(3):203-207; doi:http://dx.doi.org/ 10.3414/ME9111
27. Chen L, Hoey J, Nugent C, Cook DJ, Yu Z. Sensor-based activity recognition. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 2012;42(6):790-808
28. Kim E, Helal S, and Cook DJ. Human activity recognition and pattern discovery. IEEE Pervasive Computing 2010;9(1):48-53; doi:doi.ieeecomputersociety.org/10.1109/MPRV.2010.7
29. Krishnan N, Cook DJ. Activity recognition on streaming sensor data. Pervasive and Mobile Computing Journal; in press
30. Gu T, Chen S, Tao X, Lu J. An unsupervised approach to activity recognition and segmentation based on object-use fingerprinters. Data and Knowledge Engineering 2010;69(6):533-544; doi:10.1145/2030045.2030062
31. Lester J, Choudhury T, Kern N, Borriello G, Hannaford B. A hybrid discriminative / generative approach for modeling human activities. In: Proceedings of the International Joint Conference on Artificial Intelligence; 2005; pp 766-772; doi:10.1.1.77.5776
32. Maurer U, Smailagic A, Siewiorek D, Deisher M. Activity recognition and monitoring using multiple sensors on different body positions. In: Proceedings of the International workshop on Wearable and Implantable Body Sensor Networks; 2006; pp 113-116; doi:10.1109/BSN.2006.6
33. Rashidi P, Cook D, Holder L, Schmitter-Edgecombe, M. Discovering activities to recognize and track in a smart environment. IEEE Transactions on Knowledge and Data Engineering 2011;23(4):527-539; doi:http://doi.ieeecomputersociety.org/10.1109/TKDE. 2010.148
34. Kasteren TLM van, Englebienne G, Kröse BJA. An activity monitoring system for elderly care using generative and discriminative models. Personal and Ubiquitous Computing 2010;14(6):489-498; doi:10.1145/1409635.1409637
35. Cook DJ, Krishnan N, Wemlinger Z. Learning a taxonomy of predefined and discovered activity patterns. Journal of Ambient Intelligence and Smart Environments; in press
36. Chang C-C, Lin C-J. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2011;2(3):27
37. Hsu C-W, Lin C-J. A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks 2002;13(2):415-425
38. Korvemaker B, Greiner R. Predicting UNIX command lines: Adjusting to user patterns. In: Proceedings of the National Conference on Artificial Intelligence; 2000; pp 230-235; doi:10.1.1.35.8631