POMDP Based Action Planning and Human Error Detection

Emilie D. Jean-Baptiste, Pia Rotshtein, Martin Russell

POMDP based Action Planning and Human Error Detection

Emilie M. D. Jean-Baptiste[1], Pia Rotshtein[2], and Martin Russell[1]

School of Electronic, Electrical and Systems Engineering,[1]
School of Psychology,[2]
University of Birmingham, Birmingham, UK
{emj198,m.j.russell,p.rotshtein}@bham.ac.uk

Abstract. This paper presents a Partially Observable Markov Decision Process (POMDP) model for action planning and human errors detection, during Activities of Daily Living (ADLs). This model is integrated into a sub-component of an assistive system designed for stroke survivors; it is called the Artificial Intelligent Planning System (AIPS). Its main goal is to monitor the user's history of actions during a specific task, and to provide meaningful assistance when an error is detected in his/her sequence of actions. To do so, the AIPS must cope with the ambiguity in the outputs of the other system's components. In this paper, we first give an overview of the global assistive system where the AIPS is implemented, and explain how it interacts with the user to guide him/her during tea-making. We then define the POMDP models and the Monte Carlo Algorithm used to learn how to retrieve optimal prompts, and detect human errors under uncertainty.

Keywords: POMDP, Planning Under Uncertainty, Assistive Technology.

1 Introduction

The need to support the population suffering from cognitive impairments led to an increase number of research on assistive technology. Such systems generally focus on one specific task, and aim to provide prompts to the user in order to help him/her to complete a sequence of actions. For example, Peters et al. [1] developed the TEBRA system that supports mildly impaired people during teeth-brushing. Mihailidis et al. [2] implemented the COACH system that helps patients with Alzheimer's during hand-washing. Jean-Baptiste et al. [3] engineered an assistive system named CogWatch, which guides apraxic people during tea-making. The notion of task can also be understood as a daily plan composed of different Activities of Daily Living (ADLs). This can be seen in the Autominder project [4], which was implemented as a scheduling system for the care of the elderly, using reinforcement learning. Each system monitors the user in a sensorized environment, tracks their behaviour and infers their potential need for guidance in order to retrieve adequate prompts. Indeed, in the context of rehabilitation, guidance should be provided via errorful learning technique [5]. Hence, we consider a prompt to be adequate when (1) it is a valid continuation of what the user has achieved so far (i.e., the user state), and (2) it is retrieved because an incorrect pattern in the user's behaviour (i.e., user's error) is detected by the system. To be able to achieve this, the system relies

on an Artificial Intelligent Planning System (AIPS) that is composed of at least two modules: (1) the Action Policy Module (APM), which insures that prompts are a valid continuation of the user state, and (2) an Error Recognition Module (ERM), whose goal is to detect potential user's errors during the task. While the authors of Autominder, TEBRA and COACH all defined the probability models implemented in their AIPS's APM, no information is given about how the systems automatically detect, under uncertainty, if an error has been made by the user. Partially Observable Markov Decision Processes (POMDPs) provide a natural decision-theoric model for this problem [6, 7].

The contributions of this paper are as follows. First, we define two POMDP frameworks, and how we factorize them for ADLs management. We explain how the AIPS is trained to enable action planning via the APM, and human error detection via the ERM. Note that both APM and ERM should output their strategies based on their understanding of the user state. However, under uncertainty, the user state is represented by a belief state, which is a probability distribution over user states. Hence, the Belief State Space is infinite, and another challenge is to correctly associate a strategy to a belief state the system has not been trained for. In such a case, Nearest Neighbor Search (NNS) techniques can be applied by the AIPS to select the closest neighbor of the current belief state, and retrieve its corresponding strategy. Thus, in this paper, a specific NNS technique for ADLs management is also defined.

The paper is structured as follows. First we give an overview of the assistive system (i.e., CogWatch) in which our AIPS is implemented, then explain the technique used for its APM and ERM to fulfil their respective goals. Finally, we evaluate the POMDP-based AIPS performance via user simulation, compare it with a Markov Decision Process (MDP)-based AIPS, and discuss our future plans.

2 Assistive System Architecture

Figure 1 depicts the main modules composing CogWatch. It works as follows. First, the patient chooses the type of tea (e.g., black tea with sugar) (s)he would like to receive training for. Immersed in an instrumented environment, the patient moves sensorized objects at his/her disposal to perform actions (i.e., a_u in the figure) related to the task. When moved, these sensorized objects generate data that are passed to a module called the Action Recognition System (ARS) whose aim is to recognise the patient's actions. Processing the data in real-time, the ARS outputs an observation o that it communicates to the POMDP-based AIPS. In a realistic environment, the ARS is prone to errors, and its observation o may differ from the real action a_u made by the user. For example, when the patient adds a teabag into the cup, the ARS may believe that sugar has just been added, and pass this incorrect information to the AIPS.

Taking into account the uncertainty related to the ARS observations, the POMDP-based AIPS maintains probability distributions b_s and b_e, that it uses to select a prompt a_m^* which is composed of: (1) the best next action a_s^* the system considers the patient should follow to successfully continue the task, and (2) its best understanding of whether the patient has just made an error or not e_s^*. The AIPS prompt a_m^* is then passed to the Cue Selector, which displays, when necessary, the AIPS output in a way the patient can easily understand $\tilde{a}_m$ (e.g., still images, video, recorded message). At

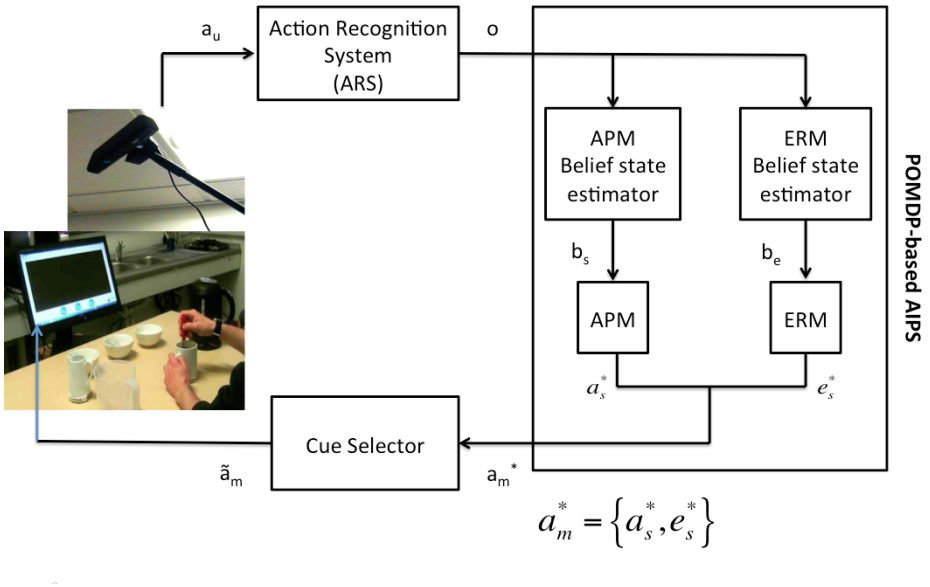

Fig. 1. CogWatch's architecture.

this point, the patient can make new actions and enter in new cycles with the system until the task is completed.

3 Overview of POMDPs

Artificial Intelligent (AI) planning has often focused on MDPs where the environment is fully observable [8]. From the AIPS point of view, the environment corresponds to the user's behaviour and his/her progression through the task when observed via the ARS outputs. Thus, when based on a MDP, the AIPS makes the assumption that there is no uncertainty related to the actions a_u made by the user; it considers that the observation o received from the ARS directly corresponds to the user's action a_u. However, as explained previously, the ARS outputs do not always correspond to the real user's behaviours. Various reasons can explain this fact, for example: (i) The sensors used to gather information about the user's environment can be noisy; (ii) Many events beyond the system's control can arise, such as exogenous events that can cause the environment to change in an unexpected way. Thus, we must consider the environment to be only partially observable. POMDPs provide a rich framework for acting optimally in partially observable domains.

Formally, a POMDP is a tuple $\{S, A, T, C, O, Z\}$ where S is a set of AIPS states; A is a set of prompts the AIPS can output; T defines a transition probability $P(s' \mid s, a_m)$;

C defines the immediate cost $c(s, a_m)$; O is a set of observations made by the ARS; Z defines an observation probability $P(o' \mid s', a_m)$.

The POMDP operates as follows. At each time step, the AIPS is in an unobservable state $s \in S$. As s is only partially known, the AIPS maintains a probability distribution over states, called the belief state b, with $b(s)$ being the probability to be in state s. Taking into account the current belief state b, the AIPS selects a prompt $a_m \in A$, then receives a cost $c(s, a_m)$ for doing so. As the ARS detects another action a_u from the user, the AIPS receives a new observation $o' \in O$, which depends on the new unobserved state s' and a_m. This new observation allows the belief state b to be updated as follows:

$$b'(s') = \frac{P(o' \mid s', a_m) \sum_{s \in S} P(s' \mid a_m, b, s) P(s \mid a_m, b)}{P(o' \mid a_m, b)} \tag{1}$$

With Equation (1), and k being a normalization constant [9], we obtain:

$$b'(s') = k \cdot P(o' \mid s', a_m) \sum_{s \in S} P(s' \mid a_m, s) b(s) \ . \tag{2}$$

4 POMDP-based APM and ERM

In this work, the output of the AIPS is the optimal prompt a^*_m, with $a^*_m = \{a^*_s, e^*_s\}$, where a^*_s and e^*_s are respectively the optimal outputs the POMDP-based APM and ERM can select. As shown in Figure 1, the modules are distinct and base their planning on different representations of the belief state. In this section, we explain how the POMDP is adapted, applied and solved, for the APM and ERM to respectively estimate their belief states b_s and b_e, and to select their respective optimal outputs a^*_s and e^*_s.

4.1 POMDP's Adaptation

CogWatch is currently used to give assistance during tea-making. When interacting with the system, the user can choose between four types of tea for which training can be given. A part of our approach consists in adapting some parameters of the POMDP to fit the system's specifications:

The Machine's Action Space A: The AIPS prompt is of the form $a^*_m = \{a^*_s, e^*_s\}$ (i.e., $a^*_m \in A$), with a^*_s being the optimal action the APM can send to the user to guide him/her during the task, and e^*_s being the optimal ERM understanding of whether the user has made an error, each time an observation is received.
Let A_s be the user's Action Space, with $A_s = \{$"Fill Kettle", "Boil Water", "Pour Kettle", "Add Teabag", "Add Sugar", "Add Milk", "Stir", "Remove Teabag", "Pour Cold Water from Jug into Mug", "Toy with the Kettle"$\}$ and E the Error Space with $E = \{$True, False$\}$. Hence, $a^*_s \in A_s$, $e^*_s \in E$, and $A = A_s \cup E$.

The Machine's State Space S: This is a finite set of machine's states s. Taking into account CogWatch's specificities, these machine's states correspond to the user states. One of our contributions is to define the machine's states as sequences of actions a_u. Thus s is defined by the tuple $\{a_u, s_d\}$, with:

- The user's action a_u (i.e., $a_u \in A_s$), being the most recent action really made by the user.
- The history of the user state s_d (i.e., $s_d \in S$), being a sequence of actions representing what the user has really achieved. For example, $s_d =$ ["Add Teabag", "Fill Kettle", "Boil Water"]). Thus, if we consider the current user state to be s_t, and the previous user state to be s_{t-1}, then $s_{t-1} = s_d$ and $s_t = s'_d$.

The Cost Function C: $C(b, a)$ is the cost for taking action a when in belief state b. As b is a probability distribution over the MDP states, we define $C(b, a)$ such as:

$$C(b, a) = \sum_{s \in S} c(s, a)b(s), \tag{3}$$

where $c(s, a)$ is the cost for taking action a when in state s. As explained in [3], $c(s, a)$ is a mechanism to incorporate human judgment about the importance of different types of behaviour into the MDP.

The Observation Space O: In CogWatch, the Observation Space is defined such as $O = A_s$. Indeed, the user's actions space A_s has been defined based on the actions the ARS can detect.

Note that, both a_u and s_d are unobservable from the AIPS perspective. Moreover, in the rest of this paper, we refer to the machine's state space as the State Space; as the latter and the user state space are the same.

4.2 Belief State Representation

In the literature, when solving ADLs with a POMDP, the POMDP belief state is generally composed of multiple probability distributions over different parameters; the latter being related to variables such as users attitude (dementia level, responsiveness, awareness), and user's behavior [10]. For COACH [11], this parametrization lead to over 22 million states; rendering explicit representation of the model infeasible. Here, we show how the belief state can directly be defined as a probability distribution over the underlying MDP state space.

Taking into account the Observation Space O, psychologists from our team defined a set of common errors cognitively impaired participants make during tea-making (e.g, sequence error, repetition error). This information is then used to restrict the State Space S, so it only contains a finite set of valid user states. We consider a user state to be valid if it does not contain any of the errors defined by psychologists. This technique allows us to maintain the MDP State Space and the belief states dimensionality at a manageable size.

From the APM's perspective The State Space S for "Black tea" contains 33 valid user states. Hence, the belief state b_s that the APM uses to plan the best next action the user should perform, is a probability distribution over 33 states.

From the ERM's perspective For the ERM to detect if the user makes mistakes during the task, we add an error state s_e in S, with s_e being an encapsulation of all user states

that psychologists consider to not be valid. Let S_e be this new State Space. In this case, the belief state b_e used by the ERM is a probability distribution over the 34 states.

Typically, both belief states can be represented such as:

$$b_s = [b_s(s_0), ..., b_s(s_{32})], with \sum_{k=0}^{32} b_s(s_k) = 1, \tag{4}$$

$$b_e = [b_e(s_0), ..., b_e(s_{32}), b_e(s_e)], with \sum_{k=0}^{32} b_e(s_k) + b_e(s_e) = 1. \tag{5}$$

4.3 Belief State Update

In the literature (e.g., COACH system), the ADL is solved using the standard definitions of the MDP or POMDP. Here, a factored POMDP is implemented. The factorization of the POMDP is a technique used in dialogue systems [12, 13]. In order to apply this technique to ADLs management, we modified the formulation of the factorization, and the formulation of the equation used to update the belief state.
Given $s = \{a_u, s_d\}$, each part of Equation (2) can be expressed by Equations (6) and (7).

Observation Function

$$P(o' \mid s', a_m) = P(o' \mid a'_u, s'_d, a_m) \approx P(o' \mid a'_u) \tag{6}$$

The observation function is the probability that the ARS makes the observation o given that the AIPS is in state s and previously selected the prompt a_m. It is simplified, taking into account the fact that, in our system, the ARS observations only depend on the actions made by the user. In this form, the observation function corresponds to the ARS confusion matrix.

Transition Probability

$$P(s' \mid a_m, s) = P(a'_u, s'_d \mid a_m, a_u, s_d) \approx P(a'_u \mid a_m) P(s'_d \mid s_d, a'_u) \tag{7}$$

The transition probability is the probability for the AIPS to be in state s', given that it was previously in s and selected prompt a_m. After decomposition of s, we can simplify it by making some assumptions:

- The probability for the user to make a new action a'_u only depends on his/her compliance to the AIPS's prompt a_m : $P(a'_u \mid a_m)$.
- The probability for the user to be in state s'_d only depends on his/her current action a_u and state s_d (s)he used to be in: $P(s'_d \mid s_d, a'_u)$.

The approximations (6) and (7), when substituted in (2), give respectively equations (8) and (9), for the APM and the ERM:

$$b'_s(s'_d) = k \cdot P(o' \mid a'_u) P(a'_u \mid a_m) \sum_{s_d \in S} P(s'_d \mid s_d, a'_u) b_s(s_d) \ , \tag{8}$$

and

$$b'_e(s'_d) = k \cdot P(o' \mid a'_u) P(a'_u \mid a_m) \sum_{s_d \in S_e} P(s'_d \mid s_d, a'_u) b_e(s_d) \ . \tag{9}$$

4.4 Offline Training and POMDPs Solving

To solve a POMDP we need to find a policy ($\pi^* : B \rightarrow A$) between the Belief State Space B and the Action Space A, such that $\pi^*(b)$ minimizes the expected cost of task completion given belief state b.
In the case of the APM, the states of the MDP are sequences of actions that can lead to successful task completion, with the Belief State Space being B_s and the Action Space A_s. In the case of the ERM, an additional state s_e is added to the MDP State Space, with the Belief State Space being B_e and the Error Space E.
Both APM and ERM are trained offline with a user simulator in order to obtain their respective policy. A Monte Carlo (MC) Algorithm [14] is used for policy optimisation. In the next sub-section, we give the general description of the optimisation algorithm implemented for our system.

Policy Optimisation During training, a belief-state-action value function $Q(b, a_m)$ records the immediate cost for taking prompt a_m in belief state b, for all belief states contained in a Belief State Space B. The Q function is defined as the expected cost of a trial starting in belief state b, the AIPS taking action a_m, and thereafter proceeding according to the current policy π until the trial ends and a final belief state is reached. During each tea trial, a sequence of belief points-action pairs $\langle\, b, a_m \,\rangle$ is recorded, and used to update the estimate $Q(b, a_m)$.
At the end of the training, when the Q values are estimated, the optimal policy π^* is obtained with:

$$\pi^*(b) = \arg\min_{a_m} Q(b, a_m), \forall b \in B, \forall a_m \in A. \tag{10}$$

The policy optimisation algorithm implemented is shown in Algorithm (1) for the APM training. Taking into account the specificities of the belief state used by the ERM, the algorithm used for its training can easily be inferred from Algorithm (1). Note that, when respectively applied to the APM and ERM, equation (10) becomes:

$$\pi^*(b) = \arg\min_{a} Q(b, a), \forall b \in B_s, \forall a \in A_s, \tag{11}$$

and

$$\varepsilon^*(b) = \arg\min_{e} Q(b, e), \forall b \in B_e, \forall e \in E. \tag{12}$$

Training via User Simulation As many trials are necessary to robustly estimate the Q values, it is common practice that a simulated user is implemented to interact with the system during policy optimisation [15, 16, 17, 18, 19, 20, 21]. Hence, we designed a Simulated User ($SimU$) based on real patients data, able to generate actions based on the type of tea selected. The $SimU$'s structure is based on data from 52 control and cognitively impaired participants, aged between 21 and 82, who completed four types of tea (black tea, black tea with sugar, white tea, white tea with sugar). Its complete architecture has already been defined in [3].

Algorithm 1 MC policy optimisation algorithm - APM

Inputs:
A: set of machine's actions a
$Q(b,a)$: expected cost for selecting a when in b
$N(b,a)$: number of times a is selected when in b
B_s : initial set of belief states b
π : initial policy, with $\pi : B_s \rightarrow A$
repeat
  $k \leftarrow 0$
  $b = b_0$ (probability 1 to be in the initial state $s_0 = \emptyset$)
  Generate a tea trial with the Simulated User:
  repeat
    $k \leftarrow k + 1$
    Get Simulated User's output $a_{u,k}$ and update b.
    $a_k \leftarrow \pi(b)$ or $\pi(b_n)$
    with $b_n \leftarrow$ nearest neighbor of b in B_s.
    record $\langle\, b_k, a_k, c(b_k, a_k)\,\rangle$, $K \leftarrow k$
  until the Simulated User terminates the trial
  Scan trial, update B and Q
  for $r \in [0, K]$ **do**
    C $\leftarrow$ sum of the costs $c(b_r, a_r)$ from b_r to b_K.
    if $\exists b_r \in B_s$ **and** $\mid b_k - b_r \mid < \epsilon$ **then**
      $Q(b_r, a_r) \leftarrow \frac{Q(b_r,a_r)*N(b_r,a_r)+C}{N(b_r,a_r)+1}$
      $N(b_r, a_r) \leftarrow N(b_r, a_r) + 1$
    else
      create a new point b_w in B_s
      initialise $Q(b_w, a_r) \leftarrow c(b_w, a_k)$
    end if
  end for
  Update the policy
  $\pi(b_r) = \arg\min_a Q(b_r, a), \forall b_r \in B_s, \forall a \in A.$
until converged

5 Evaluation

Once the policies $\pi^* : B_s \rightarrow A_s$ and $\varepsilon^* : B_e \rightarrow E$ are obtained, they are used online to evaluate the POMDP-based AIPS performance. We can define the AIPS performance as its ability to retrieve semantically optimal actions when necessary. In such a case, an action is considered to be semantically correct when it is a valid and non redundant continuation of the current user state s. It means that from the user's point of view, a_s^* should be one more step toward a successful completion of the tea (s)he is trying to make. Moreover, this prompt should only be given when necessary. This means that the AIPS should output a_m^* only when it "believes" the participant has just made an error, (i.e., when e_s^*=$True$).
Evaluating a system such as CogWatch requires many parametrizations, and is therefore complicated. Moreover, as the system is composed of distinct modules (i.e., the ARS, the AIPS, the Cue Selector); joint evaluation of the complete system is a challenge.

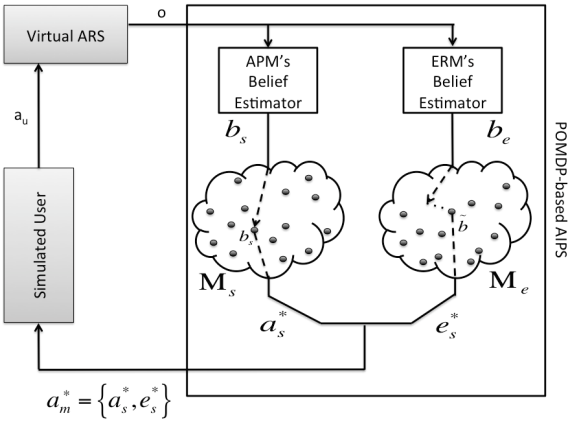

Fig. 2. Diagram of the system used to evaluate the AIPS.

Even focusing on the evaluation of the AIPS only would not decrease the complexity of the task. Indeed, the tea trials can be performed in various ways; which means that there is a vast space of user states to potentially evaluate. One approach to evaluation is to let patients interact with the system. This has been done with the MDP-based system [22]. However, it is not reasonable to have cognitively impaired patients intensively interacting with the system to solely evaluate the impact of different parameters on the system's performance. Hence, an alternative is to let a simulated user interact with the AIPS immersed in a simulation of the other components. This will enable a wider coverage of user space, and the possibility to analyse the main elements in the AIPS structure that have an impact of its ability to fulfil its goals. User simulation is a common way to evaluate POMDP techniques [16]; it is known to provide a useful basis for comparing systems [12]. The risk with using a Simulated User is that it does not properly capture real users' behaviours. However, our $SimU$ is based on real user data, and in experiments, the task completion rate for the $SimU$ and real users is similar [23].

As one can see in Figure 2, the evaluation works as follows: At the beginning of each trial (i.e., at $t = 0$), the $SimU$ is in state s_0=$\emptyset$ (the empty set). From the APM and ERM's perspective, it means that when the $SimU$ begins a trial, there is no action contained in its history; the state s_0 is empty. The $SimU$ then selects an action a_u that is passed to a virtual ARS. The latter is based on the behaviour of the real ARS. When the virtual ARS outputs an observation o, both b_s and b_e are automatically updated. The policies obtained after training are then used, allowing the AIPS to output $a_m^* = \{a_s^*, e_s^*\}$. If the belief states updated b_s and b_e are already contained in their respective policies M_s and M_e, their optimal actions are already known. On the other hand, if a belief state updated has not been visited during training and is not contained in the policy, a Nearest Neighbor Search (NNS) technique is applied to identify which neighbor should be selected (i.e., see $\widetilde{b}$ in M_e). The APM or ERM then associates the policy of this neighbor to the current belief state.

For evaluation, two main scenarios are implemented. In the first scenario, if $e_s^* = True$, a_m is passed to the $SimU$, which outputs another action a_u such as $a_u = a_s^*$; if $e_s^* = False$, a_m is not passed to the $SimU$, which outputs another action a_u based on its intrinsic settings. The AIPS performance is then calculated with:

$$P_{hits} = P_{CR} + P(a_{SC}|e_u, e_s) \ , \tag{13}$$

with P_{CR} the probability for the ERM to correctly detect that the user made no error; $P(a_{SC}|e_u, e_s)$, the probability that the APM retrieves a semantically correct action a_{SC} ($a_{SC} \in A_s$) given that the user made an error e_u and that the ERM detects that an error has been made e_s ($e_s \in E$). In the second scenario, the ERM is ignored, and the ability of the APM to correctly guide a 100% compliant $SimU$ is measured. In this case, the APM constantly tells the user what to do.

Before reporting on the POMDP-based AIPS performance, and analysing the potential impact of NNS techniques on the latter, a short overview of the NNS techniques used during evaluation is given in the next section.

6 Nearest Neighbor Search

During evaluation, the system may generate a belief state that it has never seen during training. In such a case, it will have to select a neighbor of this belief state in order to retrieve its corresponding optimal output a_m^*. This means that the choice of the metric used to select neighbors during evaluation may have an impact on the quality of the output. There are many metrics available to measure the distance between probability distributions. Some are not true metrics, but provide a notion of distance which have been proven useful to consider [24, 25]. In this paper, we will show that the AIPS performance does not only depends on the ARS error rate, but also on the metric used to select neighbors during evaluation. Hence, we will compare the impact of the correlation distance [26], euclidean distance [27] and a technique we developed and named SciMK.

SciMK measures the similarity of two belief states by comparing the underlying patterns of their MDP strategies. As explained previously, in our system, a belief state is a probability distribution over the MDP states. As each of these states can be associated to a strategy a_s^* after solving the MDP [3], each belief state can be transformed into its corresponding vector of MDP strategies. An invariant numerical label (NL) can then associated to each MDP strategy to facilitate the comparison between vectors of MDP strategies. The process is described in Figure 3:

- Step (1), the belief state is represented as a probability distribution over the MDP states.
- Step (2), the belief state is represented as a probability distribution over the MDP strategies corresponding to each MDP states from step (1).
- Step (3), the belief state is represented as a vector of numerical labels (i.e., weights), each of the latter corresponding to a specific MDP strategy.

For example, in step (3), $\pi^*(state_2) = \pi^*(state_N)$ and their corresponding numerical label is 10. It means that, based on the results found after solving the MDP, both states

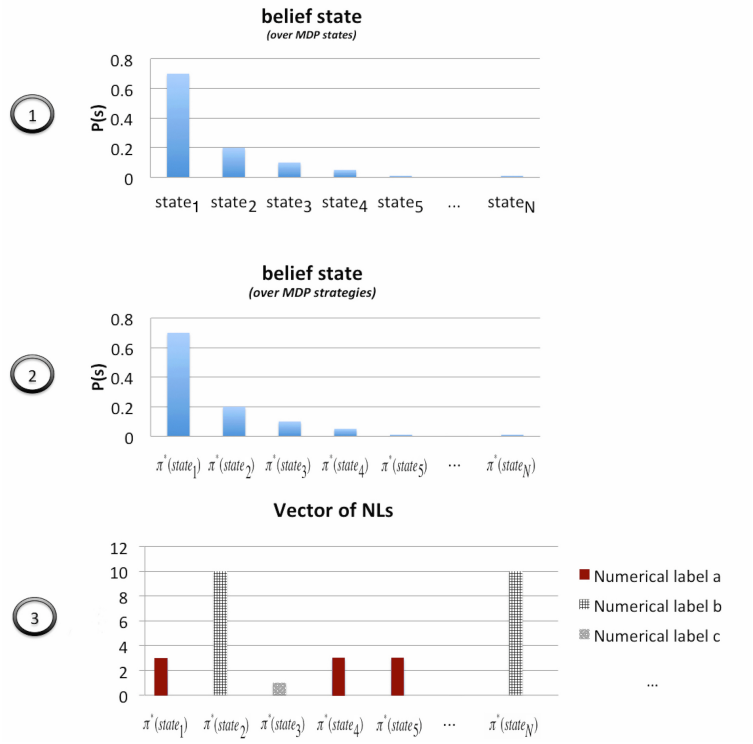

Fig. 3. Belief state transformation

$state_2$ and $state_N$ are similar to the point where the same next best action should be performed after them. Contrary to the euclidean or correlation distance, when using SciMK, we do not compare the current vector of NLs with all the elements contained in the Belief State Space. First, each belief state contained in the Belief State Space is transformed into its corresponding vector of NLs, then the Belief State Space is separated in different clusters, with each cluster being a group of vectors of NLs which had their highest probability associated to the same MDP strategy before transformation. For example, let's consider that $\pi^*(state_1) =$"add teabag". As one can see, $state_1$ has the highest probability in the belief state - step (2) in Figure 3. Thus, the vector of NLs in step (3) will only be compared to other vectors that had, in step (2), their highest probability also associated to the same strategy as the one found for $state_1$ - "add teabag". Note that after solving the MDP, different states can be related to the same strategy.

7 Results and discussion

In the first scenario, we let the $SimU$ performs the task (i.e., black tea) 300 times at varying ARS error rate. Recall that the $SimU$ complies to the AIPS's APM best next

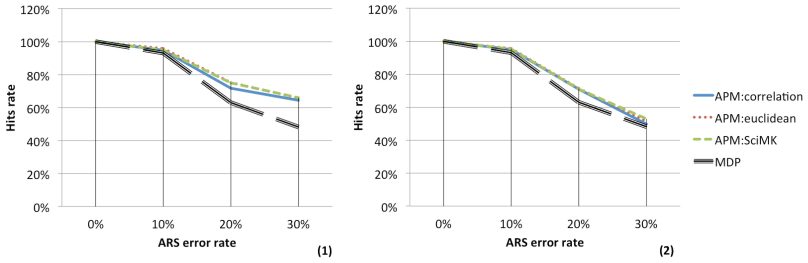

Fig. 4. Percentage of AIPS Hits at varying ARS error rate and NNS techniques. (1) ERM using correlation distance, (2) ERM using euclidean distance.

action only when the ERM detects that an error is made.
As expected, Figure 4 shows that the POMDP outperforms the MDP-based AIPS, and the ability of the AIPS (both MDP and POMDP) to output correct prompts at the right time decreases when the ARS error rate increases. Indeed, the more the ARS outputs wrong observations, the more difficult it is for the AIPS to output correct prompts. Note that at high ARS error rate (i.e., 30%), it is more advantageous to have an ERM using the correlation distance to select neighbors than the euclidean distance. On the same figure, it also seems that the percentage of hits is independent from the NNS technique used by the POMDP-based APM. However, this can be explained by our definition of P_{hits}. The latter is intrinsically dependent on the ERM ability to detect that an error has been made or not. If the ERM does not succeed to properly detect when errors really occur, there will be few occasions to measure $P(a_{SC}|e_u, e_s)$ from Equation (13). If we only focus on the ability of the ERM to correctly detect whether errors have been made or not, we can define:

$$P_{ERM_{hits}} = P_{TP} + P_{CR} \ , \tag{14}$$

with P_{TP} the probability for the POMDP-based ERM to detect that errors have really been made (True Positive), and P_{CR} same than in Equation (13). In this case, one can see in Figure 5 - (1) that the POMDP-based ERM barely outperforms the ability of the MDP-based ERM to detect errors under uncertainties. Moreover, at high ARS error rate (i.e., 30%), when the ERM uses the euclidean distance to select neighbors, the POMDP-based ERM makes more mistakes than the MDP-based one. These results highlight the fact that the POMDP framework implemented for the ERM needs to be improved. Instead of having only one state s_e that encapsulates all types of errors made, one improvement would be to extend the belief state b_e used by the ERM for it to take into account more error states. (N.B., Recall that, based on the MDP-based AIPS definition, the latter is not dependent on any NNS techniques [3].)
To have a better understanding of the APM performance only, the second scenario is implemented . In the latter, the AIPS always delivers the best next action it believes the $SimU$ should perform, whether an error has been made or not. Moreover, the $SimU$ is

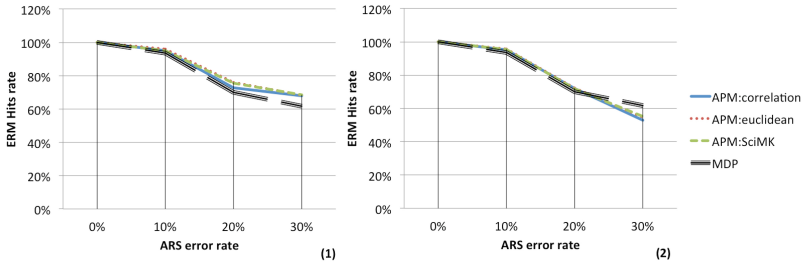

Fig. 5. Percentage of ERM Hits at varying ARS error rate and NNS techniques. (1) ERM using correlation distance, (2) ERM using euclidean distance.

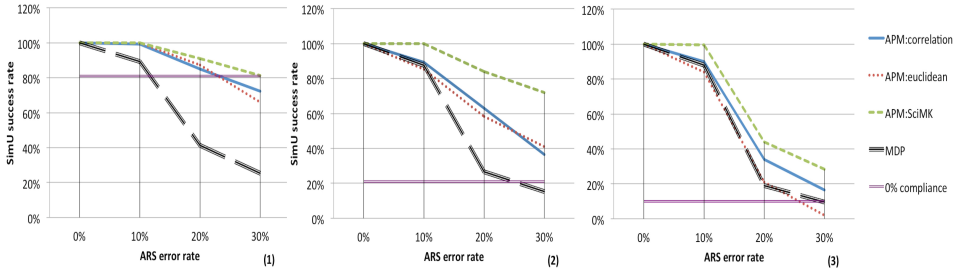

Fig. 6. SimU success rate at varying ARS error rate and NNS techniques. (1) Black tea, (2) Black tea with sugar, (3) White tea with sugar.

set to always comply to the AIPS prompts. Thus, we measured the ability of the APM to help the $SimU$ successfully complete the tasks at varying ARS error rate.

Figure 6 shows the $SimU$ success rate when performing black tea (1), black tea with sugar (2) and white tea with sugar (3). The horizontal line labeled "0% compliance" corresponds to the $SimU$ success rate when performing the task totally by itself, so when ignoring the whole system. The results show the impact of the NNS techniques applied by the APM to choose the right neighbors during evaluation. When using SciMK to select neighbors, the APM retrieves correct prompts to the user more often than when using the correlation or euclidean distance. One can also see that the POMDP-based system always outperforms the MDP's ability to guide the $SimU$; except in (3), when the APM uses the euclidean distance at 30% ARS error rate. In this specific case, it is also interesting to note that the POMDP-based APM using the euclidean distance makes the $SimU$ fails the task more often than when the latter performs it without guidance (i.e., 0% compliance). Indeed, when performing white tea with sugar without guidance, the $SimU$'s success rate is 10%, while it is close to 0% when the APM selects its neighbors with the euclidean distance, at 30% ARS error rate.

From a general point of view, Figure 6 also shows the impact of the complexity of the task on the ability of the APM (MDP and POMDP) to guide the $SimU$ properly. As explained previously, the dimensionality of the belief state b_s for black tea is 33. As this dimensionality increases with the number of mandatory steps required to complete a task, the dimensionality of b_s is 205 for black tea with sugar, and 1539 for white tea with sugar. To have a belief state being a probability distribution over 1539 MDP states means that, each time the $SimU$ selects an action, the POMDP-based APM potentially considers that the $SimU$ is in 1539 different states at the same time. The more the belief state dimensionality increases, the more complicated it is for the APM to output correct prompts. One solution to this issue would be to focus on size reduction of the MDP State Space, by merging similar MDP states based on the pattern of actions they are composed of. This dimensionality leverage of the MDP State Space will then decrease the computational burden from the POMDP-based AIPS side.

It would be interesting to compare the performance of the CogWatch AIPS with other different systems' AIPS. However, it is a very complicated task. Indeed, even if CogWatch, TEBRA and COACH focus on sequential ADLs, none shares the same POMDP framework (Action Space, State Space); all systems have been implemented to deal with different tasks; and all these tasks have been assessed in a different way by all authors; and no information is given about the NNS techniques applied by COACH or TEBRA's AIPS. Moreover, the AIPS performance highly depends on the system's ARS's error rate, which is also different in all systems. This is partially due to the fact that TEBRA, COACH and CogWatch implement their own ARS's algorithms: our system integrates a Hidden Markov Model-based action detector, TEBRA - a Bayesian network classifier, COACH - a color based flocking tracker.

8 Conclusion

In this paper, we presented the Partially Observable Markov Decision Process (POMDP) based Artificial Intelligent Planning System (AIPS) integrated in an assistive system designed for cognitively impaired participants. This system's aim is to automatically guide and assist them during Activities of Daily Living (ADLs), such as tea-making. We explained that the AIPS's goal is to provide optimal guidance, and help the user during the task when necessary. The structure of the AIPS is given, as well as the framework of its two main modules : the Error Recognition Module (ERM), whose aim is to detect human's error under uncertainty, and the Action Policy Module (APM), which has to plan optimal prompts to help the user succeed the task.
Deployed in a sensorized environment, we described how the POMDP-based AIPS copes with uncertainties, and compared its performance with a Markov Decision Process (MDP)-based AIPS in the same scenarios. We reported that the POMDP-based AIPS outperforms the MDP under uncertainty if parametrised properly. We also compared different POMDP-based AIPS with the MDP, to have a better understanding of our framework strengths and limitations. The advantage of the POMDP is its ability to model uncertainties. Each time the POMDP-based AIPS receives an observation, it updates a belief state which is a probability distribution over states. This allows the AIPS to select a prompt to guide the user that will cost the less considering all the

potential states the user might be in. However, we showed that the performance of a POMDP-based AIPS and a MDP-based one depends on the intrinsic parameters of its APM and ERM, and on the complexity of the task. Our future plan is to modify our current POMDP model in order to enable our AIPS to not only detect when the user needs assistance, but what exact type of errors that may have been made during the task. We will also work on state space leverage to improve the system's performance.

References

1. Peters, C., Hermann, T. & Wachsmuth, S. TEBRA - an automatic prompting system for persons with cognitive disabilities in brushing teeth. In *Proceedings of the 6th International Conference on Health Informatics (HealthInf)*, 12–23 (2013).
2. Mihailidis, J., Boger, J. N., Craig, T. & Hoey, J. The coach prompting system to assist older adults with dementia through handwashing: An efficacy study. *BMC Geriatrics* 8–28 (2008).
3. Jean-Baptiste, E. M. D. *et al.* Intelligent assistive system using real-time action recognition for stroke survivors. In *Proceedings of the IEEE International Conference on Healthcare Informatics (ICHI)*, 39–44 (2014).
4. Rudary, M., Singh, S. & Pollack, M. E. Adaptive cognitive orthotics: Combining reinforcement learning and constraint-based temporal reasoning. In *Proceedings of the 21st International Conference on Machine Learning (ICML 2004)* (2004).
5. Middleton, E. & Schwartz, M. Errorless learning in cognitive rehabilitation: a critical review. *Neuropsychological Rehabilitation* 138–168 (2012).
6. Astrom, K. J. Optimal control of Markov decision processes with incomplete state estimation. *Journal of Mathematical Analysis and Applications* 74–205 (1965).
7. Smallwood, R. D. & Sondik, E. J. The optimal control of partially observable Markov processes over a finite horizon. *Operations Research* 1071–1088 (1973).
8. Russell, S. & Norvig, P. *Artificial Intelligence: A modern approach* (Prentice Hall, 1995).
9. Kaelbling, L. P., Littman, M. L. & Cassandra, A. R. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 99–134 (1998).
10. Hoey, J., von Bertoldi, A., Poupart, P. & Mihailidis, A. Assisting persons with dementia during hand-washing using a partially observable Markov decision process. In *Proceedings of the Int. Conf. on Vision Systems (ICVS)* (2007).
11. Boger, J. *et al.* A planning system based on Markov Decision Processes to guide people with dementia through activities of daily living. In *Proceedings of the IEEE Transactions on Information Technology in Biomedicine* (2006).
12. Young, S. *et al.* The hidden information state model: a practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language* 150–174 (2010).
13. Williams, J. D., Poupart, P. & Young., S. J. Factored partially observable Markov decision processes for dialogue management. In *Proc. Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Int. Joint Conf. on Artificial Intelligence (IJCAI), Edinburgh* (2005).
14. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction. Adaptive Computation and Machine Learning.* (MIT Press,Cambridge, MA, 1998).

15. Levin, E., Pieraccini, R. & Eckert, W. A stochastic model of human-machine interaction for learning dialog strategies. In *Proceedings of the IEEE transactions on Speech and Audio Processing* (2000).
16. Thomson, B. & Young, S. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Computer Speech and Language* (2010).
17. Scheffler, K. & Young, S. Corpus-based dialogue simulation for automatic strategy learning and evaluation. In *Proceedings of NAACL* (2001).
18. Georgila, K., Henderson, J. & Lemon, O. Learning User Simulations for information state update dialog systems . In *Proceedings of Eurospeech* (2005).
19. Schatzmann, J., Weilhammer, K., Stuttle, M. & Young, S. A survey of statistical User Simulation techniques for reinforcement-learning of dialogue management strategies. *KER* (2006).
20. Rieser, V. & Lemon, O. Cluster-based User Simulations for learning dialogue strategies. In *Proceedings of Interspeech* (2006).
21. Quateroni, S., Gonzálex, M., Riccardi, G. & Varges, S. Combining user intention and error modelling for statistical dialogue simulators. In *Proceedings of Interspeech* (2010).
22. Jean-Baptiste, E. M. D., Howe, J., Rotshtein, P. & Russell, M. Cogwatch: Intelligent agent-based system to assist stroke survivors during tea-making. In *Proceedings of the IEEE/SAI Intelligent Systems Conference* (2015).
23. Pflugler, J. *et al.* Using human-computer interface for rehabilitation of activities of daily living (ADL) in stroke patients. lessons from the first prototype. In *Proceedings of the International Conference on NeuroRehabilitation (ICNR)*, 629–636 (2014).
24. Gibbs, A., Francis & Su, E. On choosing and bounding probability metrics. *Internat. Statist. Rev.* (2002).
25. Chung, J. K., Kannappan, P., Ng, C. & Sahoo, P. Measures of distance between probability distributions. *Journal of Mathematical Analysis and Applications* (1989).
26. Pearson, K. Notes on the history of correlation. *Biometrika* 25–45 (1920).
27. Marie, D. M. & Elena, D. *Encyclopedia of Distances* (Springer Berlin Heidelberg, 2009).