Self-Learning Algorithms for the Personalised Interaction with People with Dementia

Bram Steenwinckel and **Femke De Backere** and
Jelle Nelis and **Femke Ongenae** and **Filip De Turck**
Ghent University - imec, IDLab, Department of Information Technology,
Technologiepark 15, B-9052 Ghent
bram.steenwinckel@ugent.be

Abstract

The number of people with dementia (PwD) residing in nursing homes (NH) increases rapidly. Behavioural disturbances (BDs) such as wandering and aggressions are the main reasons to hospitalise these people. Social robots could help to resolve these BDs by performing simple interactions with the patients. This paper examines whether self-learning algorithms could be designed to select the robotic interactions, preferred by the patients, during an intervention. K-armed bandit algorithms were compared in simulated environments for single and multiple patients to find the beneficial learning agents and action selection policies. The single patient tests show the advantages of selecting actions according to an Upper Confidence Bound (UCB) policy, while the multi-patient tests analyse the benefits of using additional, contextual information. Afterwards, the learning application was provided with a framework to operate in more realistic situations. We expect that this framework can be used for personalised interactions in many different healthcare domains.

Introduction

Worldwide, almost 44 million people have dementia-related diseases, where it is the most common in Western Europ[1]. Most of these people with severe dementia are staying at nursing homes (NH), mostly specialised in dementia related dissorders (Vandervoort et al. 2013). Beside the amnesia, all these PwD suffer from so-called behavioural disturbances (BDs), like hallucinations, wandering and aggressions. Pharmacological interventions are used only for acute situations in the management of these BDs because these treatments do not address the underlying psychosocial reasons and may have adverse side effects (Fightdementia 2017). Many different non-pharmacological therapies are designed to resolve specific BDs by interacting with the PwD and without the harmful effects of medical interventions (Cohen-Mansfield 2001). Due to the increased strain on the available resources within healthcare, many NH avoid these therapies. Robots can help the nursing staff to elevate several BDs, receiving the same benefits of the non-pharmacological therapies and reduce the burden and stress of the caregivers. Many different studies investigated the effects of Robot-Assisted Therapies (RAT) onto these PwD, and neuropsychiatric symptoms tend to improve when robots are involved in simple interactions, e.g. singing a song or performing a dance (Wada et al. 2005; Valenti Soler et al. 2015; Inoue et al. 2014; Martín et al. 2013). However, these robots will only interact in a preprogrammed manner, and the nursing staff still needs to install and control these therapy sessions (Martín et al. 2013).

Robotic-assisted interventions are needed instead of the current applied therapy sessions. We are currently designing such a robotic intervention system where a care application build on top of the Nao robot could be integrated into the daily care processes for the prevention and alleviation of BDs of PwD. The necessary functionality to have such a robot autonomously walking from one resident to another, each time engaging in a personalised interaction, is currently investigated. The main idea behind these interactions is that the NAO robot will generate stimuli to elicit personal memories with associated positive feelings that have calming and reassuring effects onto these PwD.

The aim of this work is to investigate how a learning system can be built to determine which robotic action should be executed to elevate a particular BD for a specific PwD. The learning algorithms are based on reinforcement learning. A first section will give a summary of this learning technique. Taking into account this background knowledge, the second section , describes the design of the problem-specific bandit algorithms. Several simulations with virtual patients and a virtual robot were performed to investigate the performance of these bandit algorithms. The learning components were optimised and eventually surrounded by a framework to perform more practical tests, with real people and a real robot. The third section gives more details about the simulator and the designed framework. At last, the results of the tests, for different situations and both the simulated and more realistic environment, are discussed.

Background

The process, resulting in positive interactions, should learn similar like we, humans do. When we learn to ride a bike, no clear description is given of how we should perform. We just tried and, more than likely, we fell. During this learning process, the feedback signals that told us how well we did, were either pain or reward and were generated by our brain and how the environment, for instance, our parents reacted during this process. This feedback is considered reinforcement

[1] www.alzheimer.net

for doing or not doing a particular small action before receiving a much larger reward. This same technique is applicable in the field of robotics and is their more commonly known as Reinforcement Learning (RL) (Kober, Bagnell, and Peters 2015). The following section gives a brief summary about RL.

Reinforcement Learning

State, action, and reward are the three main concepts in RL. The state describes the current situation. In the case where the robot should select the most appropriate action, the state will reflect the status of the patient. An action is everything the robot can do in a particular state. When the robot executes an action in a state, it receives a reward when the intervention finishes. The term reward is a concept that describes the feedback from the environment and can be positive or negative. A positive feedback corresponds to the usual meaning of reward. When the feedback is negative, it is corresponding to what is usually called a punishment. The interaction between state, action and rewards is simple and straightforward: once the state is known, an action is chosen that, hopefully, leads to a positive reward (Sutton and Barto 2013).

The full RL paradigm, as visualised in the bottom part of Figure 1 is interested in the long-term reward after several actions were taken[2]. Every action influences the next states and different actions must be taken to receive the positive rewards. The concepts of the full RL problem can be simplified when the direct results of executing a single action gives already enough information to determine the preferred action. Another simplification is the execution of one action per intervention, which avoids the occurrence of multiple states. This simplified problem bypasses all the state concepts and learns the link between the executed actions and the possible benefits of the interactions for each state. Such a RL problem is known as a multi-armed bandit problem.

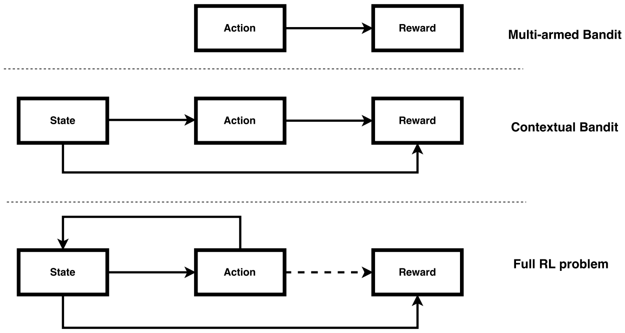


Figure 1: Top: bandit problem, where only one action effect the reward. Middle: Contextual bandit problem, where state and action effect the reward. Bottom: Full reinforcement learning problem, where action effects multiple state, and rewards may be delayed in time

(Contextual) Bandit problem When an agent has to act in only one situation, the analogy with a simple casino slot machine, where a gambler tries to maximise his or her revenue, is easily made. Consecutive handle pulls will eventually reveal some knowledge about the probability distribution of the game, and based on this gathered information, the gambler can decide which lever to pull next. Because in the end, the casino always wins, these slot machines are more commonly known as bandits. The problem of determining the patient's preferences is similar to such a bandit problem. When an intervention is needed, multiple actions can be chosen, and the goal is to get positive feedback from the patient. The probability distributions of how the patients react to these interactions are not known upfront and stay hidden inside the 'bandit' (Sutton and Barto 2013).

A k-armed bandit problem is defined by k actions and the unknown probability distribution over the rewards, $R^a(r) = \mathbb{P}[\text{r—a}]$ with a an action. At each time step t, the agent selects an action $a_t \in A$, and the reward is generated according to the selected action. The goal of a k-armed bandit algorithm is to select those actions, which maximise the cumulative reward, $\sum_{t=1}^{\tau} r_t$, with r_t the received reward in every intervention t.

Because the probability distributions of the rewards are unknown, the value for each action should be estimated. The most optimal value for this problem is denoted by v_* and follows by selecting the best possible action. The total regret score, which is the total difference of expectations between the optimal interaction and the chosen action in each time step, expresses the performance of a bandit algorithm. The bandit algorithm will have to explore the action space to find the best actions in order to calculate the total regret score.

Exploring the action space, and deciding when to exploit the current best action, is one of the main tasks of a bandit problem. Several exploration-exploitation tactics are available to balance between searching for the optimal actions and exploiting the best one. The best tactic depends on the problem. Each of these so-called policies estimates the reward value of the current chosen action in every state.

RL algorithms usually benefit from environmental information during the decision-making procedure. Many patients will react differently to the robotic interventions, and a single bandit algorithm will not differentiate between the patients. Learning the benefits of the interactions for every patient separately, for all the different situations per patient leads to a fully personalised learning system, but the learning effort grows without guaranteeing that the learning phases converges. Incorporating contextual information into the learning phase will enable the agent to learn globally over multiple patients, using some easily definable characteristics. Studies showed there is a link between the different types of dementia and the BDs (Chiu et al. 2006). Algorithms which can exploit these links are called contextual bandit problems and situations where such contextual information cannot be utilised efficiently, are rare in practice.

Bandit design

Bandit algorithms handle according to a received feedback. Such algorithms are composed of an agent, a policy and a reward signal and this section designs these three components to learn the patient's preferences.

[2] adapted from: `www.medium.com/@awjuliani`

Agent: The agent will control and execute the actions but has no capability to decide which actions should be performed. It gathers the knowledge from the environment after an intervention finishes and provides it to the learning mechanism of the bandit algorithm. The collected knowledge is usually received in the form of one or multiple reward signals. Three different types of agents were analysed.

- Normal agent: The normal agent, described by Sutton et al. (Sutton and Barto 2013), uses the reward signals to represent the feedback directly. For each action, the mean value of the unknown probability distribution will be updated with every new intervention. Based on these mean values, the algorithm can determine which action generates the highest reward.
- Gradient agent: To cope with similar rewards for certain actions, a gradient agent will try to learn the relative difference between the actions. By doing this, it can efficiently determine a preference of one action over another. The implemented gradient agent updates the preference for an action in every observation (Sutton and Barto 2013). In observation t, the agent selects action a with probability $e^{(Q[t,a]/\tau)} / \sum_a e^{(Q[t,a]/\tau)}$, where $\tau > 0$ is the temperature specifying how randomly values should be chosen and $Q[t, a]$ is the action preferences of action a at timestep t. When τ is high, the actions are chosen in almost equal amounts. As τ is reduced, the highest-valued actions are more likely to be selected and, in the limit when τ goes to zero, the best action is always chosen.
- Contextual agent: The contextual agent uses several patient characteristics to make a prediction of the probability distributions in each intervention.

Policy: Policies determine which action should be executed. They are responsible for the exploration-exploitation trade-off, which let the agent explore its actions space before exploiting the best possible interactions (Sutton and Barto 2013). Six different policies were analysed.

- Random policy: Select the actions at random for every intervention. This policy will only be beneficial when there is no task to learn, but gives a baseline solution.
- Greedy policy: At every intervention, there is at least one action whose estimated reward value is the greatest. A Greedy policy will always select this action.
- Epsilon-Greedy policy: The random policy will keep exploring and the Greedy policy starts directly exploiting. The Epsilon-Greedy algorithm combines these two polices and select a random action with probability ϵ and select the current best action with probability $1 - \epsilon$ $(0 < \epsilon < 1)$.
- Decaying Epsilon-Greedy policy: The Epsilon-Greedy policy has the disadvantage to explore forever with a predefined $\epsilon > 0$. A first phase of the learning process could need more exploration, while later on, the best actions should be exploited. It could be beneficial to start with an $\epsilon = 1$ and lower the ϵ value after every intervention until it reaches a lower bound. If this lower bound equals zero, the policy will act greedy from upon that point.
- Upper Confidence Bound (UCB) policy: Exploring the non-optimal actions according to their potential for actually being optimal, taking into account the uncertainties in those estimates, can be more beneficial. One effective way of doing this is to select actions according to the following equation:

$$a_t = \arg\max_a \left[q_t(a) + c\sqrt{\frac{\log t}{N_t(a)}} \right], \qquad (1)$$

  with $q_t(a)$ the preference of action a at timestep t, $\log_t$ the natural logarithm of t, $N_t(a)$ the number of times action a is selected at time step t and $c > 0$ the degree of exploration, defining the confidence level. The quantity being maximised is an upper bound on the possible actual preference of action a. Each time action a is selected, the uncertainty of its choice will be reduced. However, when an action different from a is selected, the uncertainty of action a increases. The UCB policy selects actions according to Equation 1.
- Contextual policy: It can be beneficial to use the predictions, based on the contextual information, to determine the action preferences. The patient's characteristics are given as input to this policy, and it outputs the expected reward for each action. The most presumable action is then selected to be executed. This policy can only be used together with a contextual agent, because the other agents do not have the predictable capacity to determine the action probabilities.

Reward: The reward signal should represent the positive or adverse effects of the executed action during an intervention. The framework operating on the Nao robot, has some useful modules to perform sentiment analyses for both vocal and visual captured data (Khosla et al. 2016). This framework can detect facial expressions, and these values were used to design a reward signal.

The facial expression analyses provides a confidence score between 0 and 1, indicating how likely an estimation is, for every of following five categories: neutral, happy, surprised, angry or sad. During an intervention, the facial expressions of the patient are analysed multiple times, and the corresponding confidence values are saved. When the intervention has finished, a service personalisation algorithm will be used to determine whether the executed action had a positive effect on the PwD. Khosla et al. (Khosla et al. 2016) designed such a service personalisation algorithm for analysing song preferences based on facial expressions using a social robot. The algorithm in this research was adapted to return an appropriate reward signal based on the captured facial expressions of an intervention. The facial expression values were divided into two groups: the first group collected all the neutral and happy facial expressions and the second group gathered the other three expression types. Equation 2 calculates the frequency of occurring expressions in both groups.

$$\tilde{f} = \begin{bmatrix} f_{\oplus} \\ f_{\ominus} \end{bmatrix} = \begin{bmatrix} n_{\oplus}/T \\ n_{\ominus}/T \end{bmatrix}, \qquad (2)$$

with $n_{\oplus}$ and $n_{\ominus}$ the number of positive respectively negative detected expressions and T the total amount of registered expressions. While the frequencies give knowledge about the occurrences of the expressions, the amount of positivity or negativity is relevant as well. The energy for the recorded expressions in both the negative and positive group is calculated using Equation 3.

$$m_e = \begin{cases} 0, & \text{if } n = 0 \\ (\sum_1^n e)/n & \text{if } n > 0 \end{cases}, \tag{3}$$

with e the values of a captured facial expression according to the associated group. Equation 4 calculates the reward signal using the energies and frequencies of both groups.

$$R = \frac{f_\oplus m_\oplus + f_\ominus m_\ominus}{f_\oplus + f_\ominus} \tag{4}$$

Application

The designed application will now combine a single agent together with a policy to learn from the developed reward signal. The policy will first select the action to be executed, and this command is sent to the robot. When the intervention finishes, the agent receives the reward and observes it by comparing its performance according to the executed action.

Various policies can be used with different agents, and many different tests are needed to find the best combinations. Testing these bandit algorithms directly onto PwD could result in stress and are therefore avoided. A simulated application was designed to mimic the behaviour of these people, and tests were executed with this simulator. How a PwD reacts to a robotic action is, however, unknown. The simulator analysed three different reaction strategies. One strategy defines a single action which had a highly positive effect onto the PwD in comparison with the other actions. Another strategy has four similar action effects and the last strategy defines four actions having an adverse effect onto the PwD, but with one action slightly less bad. These strategies resemble an optimistic, neutral and worst case scenario respectively. Several tests compared these three different cases.

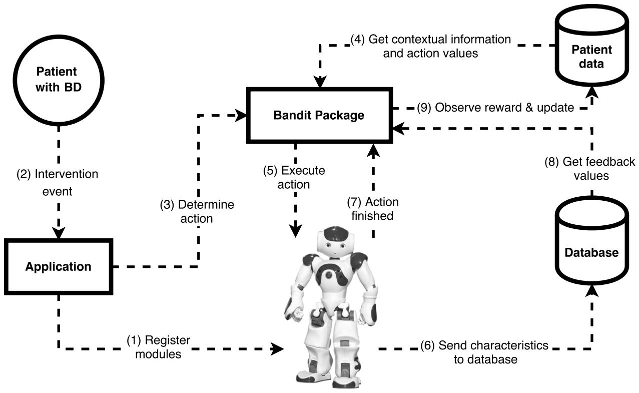


Figure 2: Robotic application overview

Figure 2 shows the designed application. The application aims to react upon BD events. The application will then first register several modules from the Nao robot to enable the communication (1). When an intervention is needed, the application is signalised using such events (2). The robot executes (5) the chosen (3-4) action, and facial expression data is sent to a database during the intervention for further analyses (6). When the intervention finishes (7), the bandit algorithm will gather all this stored data and builds a reward signal (8). The agent observes this feedback, and preferences are updated for further interventions (9). The most promising algorithms from the simulated tests were used in a more realistic setting. During these practical tests, a real person mimicked expressions in front of a Nao robot.

Results

Three different test approaches were used to investigate, whether the designed bandit algorithms could be used for learning the action preferences of PwD. A first test case examines the effects of bandit algorithm onto a single patient in a simulated environment. The second test case analyses the bandit algorithms onto multiple patients, again in a simulated environment using contextual information. The last test case uses the designed application in Figure 2 to test the action preferences of two real persons, using a real robot.

Single patient simulations

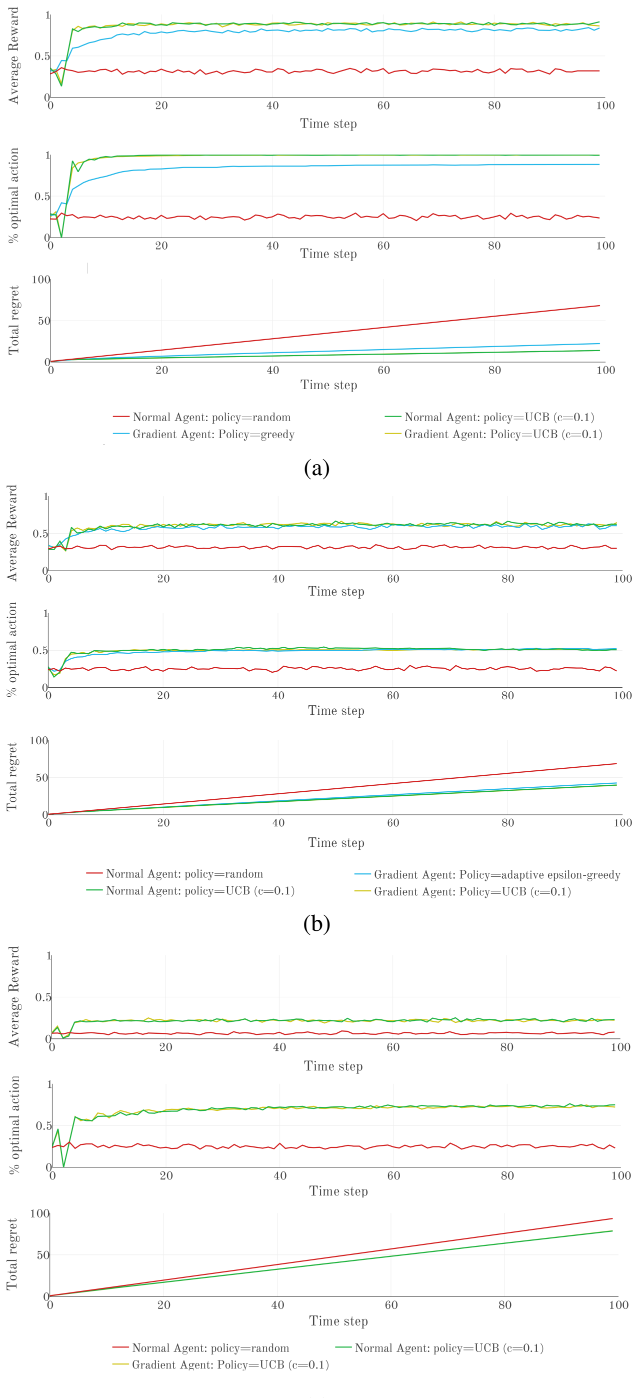
The normal and gradient agents are compared, together with the five possible policies. Results are visualised using three different plots describing the average amount of rewards the bandit algorithms received over multiple interventions, the number of optimal actions selected during the interventions and the cumulative regret score over multiple interventions.

Tests for an optimistic case were executed for 20 randomly generated virtual patients and 4 actions could influence the mood of a virtual patient during 100 consecutive interventions. The results in Figure 3 (a) show various bandit algorithms after performing 100 such experiments where the random policy act as a baseline for the other bandit algorithms. The UCB policy can learn the preferred action after 4 interventions and has a low optimal regret score. A Greedy policy in combination with a gradient agent has almost the same average reward values, but selects frequently sub-optimal actions.

The same tests were executed in a neutral (3 (b)) and worst case situation (3 (c)). The similarity between the actions changes the action selection strategy. The algorithms are still better than the random policy which selects the best action one out of four. The average reward values are very low in the worst case situation. Despite, the most bandit algorithms succeeded in detecting the preferred actions.

Multi-patient simulations

During the multi-patient tests, contextual information from the virtual patients can be used to determine the action preferences. These tests will try to learn the action preferences for multiple patients. In each intervention, the virtual patient can now be different. These tests examine the benefits of using contextual information instead of learning the action preferences from multiple PwD separately for each patient. Figure 4 summarises the results for 100 experiments



(a)

(b)

(c)

Figure 3: Overview of the bandit algorithm tests for the (a) optimistic, (b) neutral and (c) worst case when a single patient is considered.

with each 200 consecutive interventions. Gradient and normal agents are still individualised for all these tested patients and need more time to reach the global optimality. A contextual agent benefits clearly from the additional available information. The optimal actions are selected more frequently over the whole duration of these tests. Similar tests for the neutral and worst case scenario were executed and behaved similarly to the single patient results.

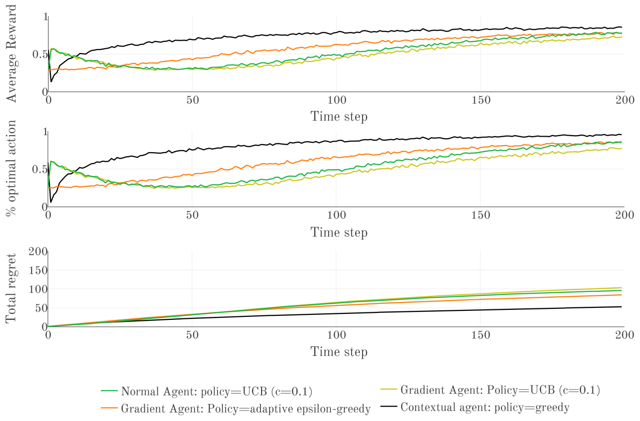


Figure 4: Overview of the bandit algorithm tests for the optimistic case when multiple patients are considered.

Behavioural patterns of patients often change. Figure 5 shows the results of the test when the behavioural pattern of multiple virtual patients changes randomly in the 200th intervention. All the bandit algorithms can detect this change, but the contextual agent using a contextual policy can recover easily from such situations using the additional contextual information.

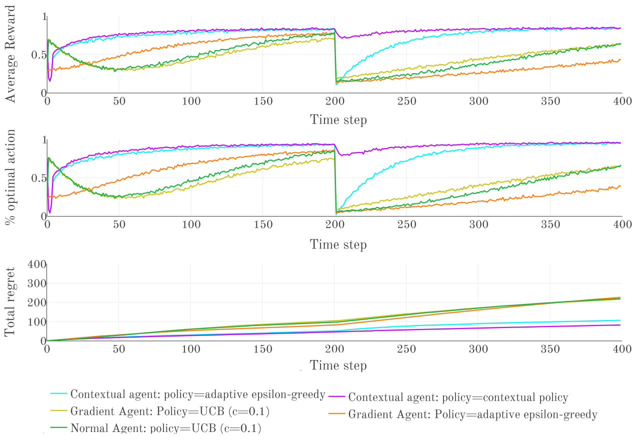


Figure 5: Overview of the bandit algorithm tests when a change in behaviour occurs for multiple patients during the 200th timestep.

Real robotic test

Results for the tests, which analysed the correct functioning of the designed application using a real Nao robot and with a real 24-year old person, are shown in Figure 6. The person, without a dementia related disease, mimicked several facial expressions in front of the robot which announced actions according to a UCB policy, in combination with the gradient agent.The robot could easily detect the action preferences after 20 interventions and the UCB policy could easily differentiate between the small differences and exploit the optimal action after a limited number of interventions. When the certainty of an optimal action lowers due to less convinced expressions, the policy will shift its selection procedure.

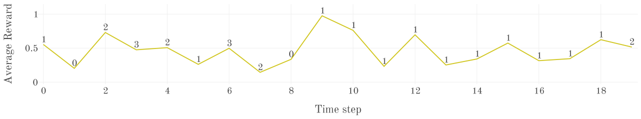


Figure 6: Robotic test with 20 interventions and an UCB policy with c=0.01 for the test person (duration single event: 30s).

Conclusion

This paper investigated whether self-learning algorithms could be designed for the personalised interaction with PwD when a particular BD occurs and needs to be alleviated. RL techniques were used to determine the action preferences of the patients and selected the one which resulted in the highest reward. The less complex bandit algorithms can be applied when different state occurrences were ignored. Several different bandit policies were compared to find the best balance between exploring and exploiting the action space. The facial expressions of the patients, analysed by the Nao robot, were used to provide a feedback signal for these bandits. Tests in a simulated environment with a single virtual patient showed the advantages of using an UCB policy which could determine the preferred actions quickly. During multi-patient tests in the same simulated environment, the patient-specific information gives some clear benefits for learning globally. Therefore, the contextual agent using a contextual policy can recover fast from changes in behavioural patterns. In a more realistic setting, with a real robot and real people, tests showed the correct functioning of both the learning algorithm and the designed framework. Further tests on real PwD will be needed to optimise the developed learning application. The interactions between the robot and the patient could also be used to detect changes in behaviour, by searching for additional links between the patient's expressions and his or her contextual information.

Acknowledgement

This research is funded by the imec.icon project WONDER, which aims to develop an automated and interactive 24/7 care service with a humanoid care robot[3] and brings together the following partners: QBMT, Xetal, WZC Weverbos, WZC De Vijvers, CM, Ghent University, SMIT and iLab.t.

References

Chiu, M.-J.; Chen, T.-F.; Yip, P.-K.; Hua, M.-S.; and Tang, L.-Y. 2006. Behavioral and Psychologic Symptoms in Different Types of Dementia. *Journal of the Formosan Medical Association* 105(7):556–562.

Cohen-Mansfield, J. 2001. Nonpharmacologic Interventions for Inappropriate Behaviors in Dementia. *Am J Geriatr PsychiatryAm J Geriatr Psychiatry* 94(9):361–381.

Fightdementia. 2017. Alzheimer's Australia — Drugs used to relieve behavioural & psychological symptoms of dementia.

Inoue, K.; Sakuma, N.; Okada, M.; Sasaki, C.; Nakamura, M.; and Wada, K. 2014. Effective application of PALRO: A humanoid type robot for people with dementia. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8547 LNCS, 451–454.

Khosla, R.; Nguyen, K.; Chu, M.-T.; and Tan, Y.-A. 2016. Robot Enabled Service Personalisation Based On Emotion Feedback. *Proceedings of the 14th International Conference on Advances in Mobile Computing and Multi Media - MoMM '16* 115–119.

Kober, J.; Bagnell, J. A.; and Peters, J. 2015. Reinforcement learning in robotics :. *The International Journal of Robotics Research* 32(11):1238–1274.

Martín, F.; Agüero, C. E.; Cañas, J. M.; Valenti, M.; and Martínez-Martín, P. 2013. Robotherapy with dementia patients. *International Journal of Advanced Robotic Systems*.

Sutton, R. S., and Barto, A. G. 2013. *Reinforcement learning : an introduction*, volume 9. A Bradford Book.

Valenti Soler, M.; Agüera-Ortiz, L.; Olazaran Rodriguez, J.; Mendoza Rebolledo, C.; Pérez Muñoz, A.; Rodriguez Pérez, I.; Osa Ruiz, E.; Barrios Sanchez, A.; Herrero Cano, V.; Carrasco Chillon, L.; Felipe Ruiz, S.; Lopez Alvarez, J.; Leon Salas, B.; Cañas Plaza, J. M.; Martin Rico, F.; and Marti, Martinez, P. 2015. Social robots in advanced dementia. *Frontiers in Aging Neuroscience* 7(JUN).

Vandervoort, A.; Van den Block, L.; van der Steen, J. T.; Volicer, L.; Stichele, R. V.; Houttekier, D.; and Deliens, L. 2013. Nursing Home Residents Dying With Dementia in Flanders, Belgium: ANationwide Postmortem Study on Clinical Characteristics and Quality of Dying. *Journal of the American Medical Directors Association* 14(7):485–492.

Wada, K.; Shibatal, T.; Musha, T.; and Kimura, S. 2005. Effects of robot therapy for demented patients evaluated by EEG. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*.

[3] www.imec-int.com/en/what-we-offer/