Exploring Clustering Techniques for Effective Reinforcement Learning based Personalization for Health and Wellbeing

Eoin Martino Grua
Department of Computer Science
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
e.m.grua@vu.nl

Mark Hoogendoorn
Department of Computer Science
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
m.hoogendoorn@vu.nl

***Abstract*—Personalisation has become omnipresent in society. For the domain of health and wellbeing such personalisation can contribute to better interventions and improved health states of users. In order for personalisation to be effective in this domain, it needs to be performed quickly and with minimal impact on the users. Reinforcement learning is one of the techniques that can be used to establish such personalisation, but it is not known to be very fast at learning. Cluster-based reinforcement learning has been proposed to improve the learning speed. Here, users who show similar behaviour are clustered and one policy is learned for each individual cluster. An important factor in this effort is the method used for clustering, which has the potential to influence the benefit of such an approach. In this paper, we propose three distance metrics based on the state of the users (Euclidean distance, Dynamic Time Warping, and high-level features) and apply different clustering techniques given these distance metrics to study their impact on the overall performance. We evaluate the different methods in a simulator with users spawned from very distinct user profiles as well as overlapping user profiles. The results show that clustering configurations using high-level features significantly outperform regular reinforcement learning without clustering (which either learn one policy for all or one policy per individual).**

***Index Terms*—Health care, Reinforcement Learning, Personalization**

I. Introduction

Personalisation is defined by [1] as "a process that changes the functionality, interface, information access and content, or distinctiveness of a system to increase its personal relevance to an individual or a category of individuals". Personalisation has become omnipresent in our society (e.g. [2]–[5]). While applications were historically limited to web shops and alike, a whole range of applications can nowadays be seen.

What technique is best suited to obtain personalisation depends greatly on the task at hand. Take personalisation for health and wellbeing. In such a setting one aims to perform actions to influence the behaviour and physical state of the user to improve the overall health state. The health setting is challenging: consequences and appropriateness of actions cannot be observed immediately. Some actions might have a negative impact at first, only showing benefit in the distant future. In addition, the appropriateness of actions is likely very dependent on the user context. One technique which can be used for personalisation fits this setting very well is reinforcement learning (cf. [5]). Unfortunately it does have its downsides: the learning process can be very slow (requiring a lot of experiences) and exploring undesired or ineffective parts of the action space can lead to user disengagement.

Several approaches have been proposed to overcome these problems. One set of approaches includes the usage of transfer learning, i.e. reusing previously generated policies (cf. [6]). Alternatively, [7] have proposed to cluster users to make the reinforcement learning process more effective while still enabling a level of personalisation. In the case of [8], users are assigned to a cluster after some initial period, and a policy is learned per cluster. While the initial results are promising, the results highly depend on the quality of the clustering (cf. [8]), i.e. whether the users in a cluster are sufficiently alike in terms of the policy that works best for them.

In this paper, we explore cluster-based reinforcement learning more in depth, focusing on the approach to cluster users. We define different distance metrics based on the states of the users (based on the Euclidean distance, Dynamic Time Warping cf. [9], and by deriving high-level features), and combine them with two well-known clustering techniques (Agglomerative Clustering and K-Medoids). Next, we study the influence of the choice upon the overall performance in terms of personalisation. In addition, we investigate how the presence or absence of very distinct groups of users impacts the benefit of using cluster-based reinforcement learning. We make use of an existing simulation environment [8] which allows the simulation of users in a health context (focused on getting people to perform sufficient daily physical exercise). Using such a simulator allows us to easily manipulate users, their behaviour and the existence of distinct profiles, hence, it allows us to purely focus on the clustering techniques themselves.

This paper is organized as follows. First, we will describe related work in Section 2. Section 3 details our proposed clustering approach, while Section 4 briefly describes the simulator we use for our experiments. The experimental setup

is described in Section 5 and the results in Section 6.

II. Related Work

As discussed in the introduction we use reinforcement learning as a mean to learn when to give the intervention to the user (in our case the generated agent). Reinforcement learning has not been applied frequently in health intervention settings while it is well suited for these types of problems (see e.g. [10], [11]). There are however some papers that have already explored its suitability.

[12] proposed the use of reinforcement learning to help decide on the correct type of message needed to be sent to users of a mobile application affected with diabetes type 2 to encourage physical activity. The role of the reinforcement learner was to correctly choose the type of message that would most effectively encourage the patient to increase his/her physical activity (which is beneficial for patients with diabetes type 2). This case is an example of a one-size fits all model.

[7] addresses the problem with using either a one-size fits all policy and using individual learning. They suggest the use of clustering to achieve a balance between the amount of data available to the learner and the individual personalisation. They show that with the cluster-based reinforcement learning, they manage to achieve higher values of reward compared to both other methods, though they assumed a fixed clustering approach and the action space was limited.

Whilst the previous studies have commonalities with our work, the most similar study is [8]. Here the authors expand on the work of [7] and built a dedicated simulator to evaluate the approach for more difficult scenarios. That same simulator is used in our study. Furthermore, we wish to employ the setting used by [8] whilst expanding the clustering analysis component.

Lastly, our work also contains similarities to transfer learning [6] where a learned policy from one task can be transferred to another, which in our case could apply to the use of the learned policy from one user (or group of users) to a new user. This is not done in our particular study due to the assumption of a universal timeline for all agents generated.

III. Approach

As explained before, we exploit cluster-based reinforcement learning to improve the learning speed of reinforcement learning algorithms in a health and wellbeing context. Here, we focus on learning how to provide the most effective interventions to improve the future health state of the user. Our precise case study will be explained in the next section. In this section, we focus on the reinforcement learning component first. As a starting point, we formulate the problem. We will use a model-free reinforcement learning formulation. After we have defined this formally, we will focus on learning reinforcement learning policies for users. Then we will go to the main contribution of this paper, namely the introduction of different clustering approaches to cluster users and learn policies over such clusters to improve the learning speed and quality.

A. Reinforcement Learning Problem Formulation

The problem we are facing is a control problem, which we model using a Markov Decision Process (MDP) [10]. This formulation follows (cf. [8]). In our formulation, we identify a user with the subscript u (with $u \in U$). The MDP for our problem can be specified as $M_u = \langle S_u, I, T_u, R_u \rangle$. Here, S_u specifies the user states, and I represents the interventions that can be selected (i.e. actions in reinforcement learning terms). T_u specifies the probabilistic transition function of a user u and is defined as follows $T_u :: S_u \times I \times S_u \rightarrow [0, 1]$. This function expresses the probability of moving from one user state to another, provided that we have selected an intervention from I. R_u is the reward function, which assigns a reward based on the observed state s_u and the intervention $i \in I$ provided to user u. Since we are dealing with human subjects in our setting, we cannot assume complete knowledge. T_u cannot be directly accessed (i.e. we assume it to be unknown). Furthermore, we cannot observe the full state, but only a vector of features ϕ derived from the state $s_u \in S_u$. Considering p features we specify this vector as follows: $\phi(s_u) = \langle \phi_1(s_u), \ldots, \phi_p(s_u) \rangle$. While we cannot know up front whether the process in fact satisfies the Markov property, we assume the process to be sufficiently close such that we can employ standard reinforcement learning algorithms.

Given this problem formulation, we want to learn a policy π_u per user, that expresses what intervention should be selected in which state $\pi :: S_u \rightarrow I$. Applying such a policy results in experiences for each time point t: $\langle \phi(s_u^t), r_u^t, i^t \rangle$. Here, we use t to identify the specific time point. These experiences together accumulate in traces (referred to as Σ) for each user u: Σ_u (with T being the last time point):

$$\langle \phi(s_u^t), r_u^t, i^t, \phi(s_u^{t+1}), r_u^{t+1}, i^{t+1}, \ldots, \phi(s_u^T), r_u^T, i^T \rangle \quad (1)$$

We define the value of doing intervention i in state s as:

$$Q^\pi(s, i) = E_\pi \{ \sum_{k=0}^{\infty} \gamma^k r^{t+k+1} | s^t = s, i^t = i \} \quad (2)$$

γ is a discount factor for future rewards. Then, the policy we strive to find maximizes this value (i.e. selects the best interventions in each state):

$$\pi'(s) = \arg\max_i Q^\pi(s, i), \ \forall s \in S \quad (3)$$

To find such a policy, we deploy an off-policy reinforcement learning algorithm, namely Least Square Policy Iteration (LSPI) [13]. This uses the feature vector of the state ($\phi(s)$) and finds a linear approximation of the Q function by means of a weight vector $\langle w_1, \ldots, w_p \rangle$ containing a weight for each of our p features from a batch of experiences. Different alternatives are possible, but this is outside the scope of this paper. The techniques explained below are however independent of the specific reinforcement learning algorithm that is selected.

B. Learning Policies

One of the problems when dealing with human users is that there is hardly room for an exploratory phase in which a lot of different actions can be tried. Furthermore, the state space (even when using our feature vector ϕ) is potentially very large. When we learn our policy, we can make a choice how user specific we want to learning such a policy. We can:

- learn one policy over all users (Pooled approach)
- learn one policy per user (Separate approach)
- learn one policy per group of similar users (Clustering approach)

The first two options are simple. For learning, we can simply vary what experiences we feed to our reinforcement learning algorithm. For learning one policy over all users, we provide $\Sigma = \{\Sigma_u | u \in U\}$ and generate a single policy across all users. For learning a policy for a single user, we only provide the experience for that user: $\Sigma = \{\Sigma_u\}$. Both options come with downsides. Learning one policy across all users will highly likely result in insufficiently tailored interventions, while learning per individual will suffer from a lack of experiences to learn a reasonable policy in a short time frame. We therefore study learning across groups of users that seem to be relatively alike (following [8]). We define these groups using clustering techniques, and want to learn policies per cluster. We provide the learning algorithm with the following experiences: $\Sigma = \{\Sigma_u | u \in C\}$. While learning across such clusters has already shown to be beneficial (cf. [8]), the impact of the clustering approach itself has not been studied in depth.

C. Clustering

In order to define clusters, we need to have (1) a clustering technique, and (2) a distance metric. Let us consider the distance metric first. We will refer to the distance between a user u_1 and user u_2 as $d(u_1, u_2)$. What can we base this distance metric on? Initially, we assume to have no knowledge about the specific users (and hence, we cannot determine a distance between users). We therefore start with a so-called *warm-up phase* where we gather experiences of users with a random policy. Once collected, we can define a distance between the experiences we have gathered for the users. These experiences are in fact temporal sequences of the information we have available about the user at each time point (the features describing the state of the user, the intervention, and reward information). We define three distance metrics between experiences of users: (1) using the Euclidean distance, (2) using Dynamic Time Warping (cf. [8]), and (3) using derived features.

For the *Euclidean distance*, we measure the distance between the states of the user, and do not consider the actions and rewards. The rational behind our decision on only including the states is because the states are a closer representation of the behaviour of the agent as defined by the profile settings. We did not want to include information that is more dependant on the setup of the learner in the clustering of the agents. We assume that the feature vector ϕ (representing what we observe about the state f the user) only contains numerical features. If there are categorical features, we can encode categorical features using one hot encoding. To calculate the distance we simply compare the difference between the state features as follows:

$$d_{ED}(u_1, u_2) = \sum_{t=0}^{T} \sqrt{\sum_{i=1}^{p} (\phi_i(s_{u_1}^t) - \phi_i(s_{u_1}^t))^2} \tag{4}$$

In this calculation, we assume that the sequences of both users are of equal length and their start times have been synchronized. The second approach considers *Dynamic Time Warping (DTW)* [9]. This allows for a more flexible matching between the experiences of users, where the speed of the sequences might be different. As a basic building block, a distance function between two experiences of users is defined:

$$d_{ED}(u_1^t, u_2^{t'}) = \sqrt{\sum_{i=1}^{p} (\phi_i(s_{u_1}^t) - \phi_i(s_{u_1}^{t'}))^2} \tag{5}$$

Again, we only consider distances between the features of the states. DTW tries to match time points in order to minimize the sum of the distances over time points provided that: (1) the first and last time points of both sequences are matched, and (2) a monotonicity condition is satisfied. See [9] for more details. For the DTW, we split the sets of experiences into a number of intervals of k discrete time points within which we perform the DTW (i.e. $[t, \ldots, (t+k)), \ldots [(T-k), T)$). For example, think of splitting the sequences of experiences into days, and comparing how equal the states within a day are. This is done for computational reasons, but also because we do not want to match outside of these boundaries to avoid overly optimistic matches over days. The overall user distance is defined as:

$$\begin{aligned} d_{DTW}(u_1, u_2) = \sum_{i=0}^{T/k} dtw(&\langle \phi(s_{u_1}^{i \cdot k}), \ldots, \phi(s_{u_1}^{i \cdot k + (k-1)}) \rangle, \\ &\langle \phi(s_{u_2}^{i \cdot k}), \ldots, \phi(s_{u_2}^{i \cdot k + (k-1)}) \rangle) \end{aligned} \tag{6}$$

The final distance metric we consider is *derived features* from the sequences of experiences and comparing on that higher level. An example is to derive the average values per feature over the entire series of experiences and compute the distance between those averages:

$$d_{DF}(u_1, u_2) = \sqrt{\sum_{i=1}^{p} \left(\frac{\sum_{t=0}^{T} \phi_i(s_{u_1}^t)}{|\{0, \ldots, T\}|} - \frac{\sum_{t=0}^{T} \phi_i(s_{u_2}^t)}{|\{0, \ldots, T\}|} \right)^2} \tag{7}$$

Given these distance metrics, we can apply standard clustering techniques (which we deliberately leave open in this approach). These are commonly parameterized algorithms, which require a selection of the number of cluster (e.g. k in K-Medoids clustering) or a threshold to be set which in

fact determines the number of clusters (e.g. in Hierarchical Clustering). To select the best value for a parameter, we use an evaluation metric commonly used in clustering to evaluate the quality of the clusters: the silhouette (cf. [14]). We run the clustering algorithms for various parameter settings and select the setting which results in the highest quality clustering with this metric.

IV. Simulator

To test our approach, we utilize a simulation environment[1] which is able to generate realistic behaviour of human-like agents for a health and wellbeing setting. The simulator we use is described more extensively in [8]. It focuses on trying to coach people towards a healthier lifestyle by engaging them more in sports, a common goal among health apps available in the iTunes or Google Play store [15]. The simulator emulates the behavior of human beings by generating their activities throughout the day (e.g. working, eating, working out) as well as their responses to interventions they receive in the form of messages that encourage them to work out. How schedules and responses are generated is based on certain generic profiles (e.g. think of an average working person). States are observed once per hour. The features of the state (ϕ) are the current day of the week, the current hour of the day, if the agent has worked out within the current day, the fatigue level and which of the possible activities he performed in the captured hour.

As said, the acceptance of the intervention depends on the schedule of the agent, their fatigue level, as well as their profile. The reward is given based on a few conditions. If the agent accepts the intervention given, a reward of +1 is recorded, whilst if the intervention is rejected then a negative reward of -0.5 is returned. If the intervention was accepted an extra reward of +10 is given when the workout is completed. The duration of the workout can also be considered but for our setup we have decided not to do so. The final condition that can score reward is the level of fatigue of the agent. The amount of negative reward recorded increases with the amount of fatigue. For our case fatigue is defined as an incremental integer that starts from 0 and increases for every consecutive workout. The moment the agent skips, rejects or is not told to workout the fatigue level is reset to 0. As an example, if an agent works out three days in a row (each day working out once) its current fatigue level is equal to three. When the fourth day the agent does not workout the fatigue level gets reset.

For our investigation, we use sets of three profiles from which agents are spawned. The technicalities of each profile used are explained in subsection V-C. The simulator has been implemented in Python3.

V. Experimental Setup

In order to evaluate our approach, we perform a number of experiments. In this section, we explain the different experimental conditions, the performance evaluation, and the parameters and simulator settings.

A. Experimental Conditions

We are interested in studying the performance of our cluster based learning approach compared to the two alternative variations we mentioned in Section III-B (pooled and separate). In addition, we want to understand how the distance metric and the selected clustering algorithm impacts performance. We use our three distance functions and combine these with two commonly known clustering algorithms, namely K-Medoids clustering [16] and Hierarchical Clustering (Agglomerative Clustering, using the complete linkage criterion) [17]. While more advanced clustering algorithms are available, we want to start with relatively simple approaches which can also easily be combined with the various distance functions chosen. Overall, this results in $2 \times 3 = 6$ variations for the clustering. Thus we have 8 variations of the reinforcement learning algorithm in total.

How easily groups of users can be distinguished (and whether they are present or not) is likely to have a severe impact on the advantage of using a cluster-based approach. To study this influence, we try two different setups of our simulation environment. One setup features three highly distinctive profiles (both in terms of their daily schedules and responses to the received interventions) while the second setup will again be three profiles but with two being very difficult to distinguish. Subsection V-C shows the specification of the profiles used in both settings.

B. Performance Evaluation

To evaluate the performance of the algorithms, we focus on two aspects.

To study the performance of the clustering itself, we apply clustering to the traces of experiences we collect during the *warm-up phase* in which we apply a random policy. We study the users residing in the resulting clusters and consider the original profiles they were spawned from. A desirable outcome would be to see low diversity of profiles within a single cluster. We perform five runs per clustering algorithm as the results are highly dependent on the random initialization of the centres (certainly for K-Medoids).

The second evaluation is the performance of the reinforcement learning algorithm and the resulting reward. Hereto, we consider the average reward we obtain. Next to the aforementioned *warm-up* period, we apply a *learning period* during which we measure the reward. For all variants, after the *warm-up* days we create a policy using LSPI and train each LSPI instance over the traces of the associated agents. Each policy is then updated on a daily basis over the remaining *learning period* and used to select the interventions. We compute the average daily rewards over all runs, agents and time points per day (called the average daily reward).

The best performing clustering configurations will be selected and compared to both the *separate* and the *pooled* cases. To determine whether the difference between trends is

[1] 'https://github.com/EMGrua/MultiAgentSimulation-MultiClusterVariation'

statistically significant we used the Wilcoxon signed-rank test. We define various levels of significance: one star ($\star$: $P \leq 0.05$); 2 stars ($\star\star$: $P \leq 0.01$), and three stars ($\star\star\star$: $P \leq 0.001$).

C. Parameter and Simulator Settings

For each experiment the simulation was ran with a constant set of parameters. These parameters were chosen based on preliminary experiments and feasibility of the run times. The parameters chosen were:

- *Number of agents*: the number of agents for all runs was set to 100, with the agent profiles being equally distributed among them, so we always expect a profile distribution of 33-33-34.
- *Warm-up phase*: the 'warm-up phase' was set for all runs to 7 days.
- *Learning phase*: the 'learning phase' was set to 60 days (which is enough to obtain a stable policy).

The simulation parameters that were changed according to the executed experiment were the profile types. Below we list the two sets used (distinct and overlapping) as well as the key differences between each type of profile. The *distinct profiles* are:

- *Worker*: works 5 days a week plus he has a 80% of working on the sixth day (Saturday). The Worker starts anywhere from 8 a.m. to 9 a.m. and works for 10 to 11 hours. Gets fatigued after 2 consecutive workouts and has a 10% chance of accepting a second workout in the same day.
- *Athlete*: works 3 days a week (Monday, Tuesday and Thursday) starting from around 9 a.m. for 8 hours. The athlete gets fatigued after 4 consecutive workouts and has a 50% chance of accepting a second workout in the same day.
- *Retired*: never works. The retiree gets fatigued after one workout and will never accept a second workout on the same day.

The *overlapping profiles* are:

- *newWorker*: identical to Worker but does not have a chance of working a sixth day. The newWorker is also identical in the way it behaves with working out and fatigue pattern.
- *newAthlete*: identical to the athlete but has a 60% chance of working on Wednesday and a equal chance of working on Friday. NewAthlete is also identical to Athlete in the fatigue and workout settings.
- *Athlete*: identical to the previously described athlete.

It is important to remember that apart from these differences all of the profiles include routine actions, such as eating (breakfast, lunch, dinner) and sleeping.

VI. Results

In this section, we present the results we obtained using the experimental setup we have just described. We start with the analysis of the clusters, followed by the performance of the reinforcement learning techniques[2].

[2]The data can be found here:'http://doi.org/10.5281/zenodo.1215905'

TABLE I
NUMBER OF CLUSTERS RETURNED BY EACH EXPERIMENTAL CASE (FOR THE DISTINCT PROFILE CASE)

	run1	**run2**	**run3**	**run4**	**run5**	**Mode**	**Median**
keu	2	2	3	3	4	2,3	3
heu	6	4	6	2	3	6	4
kdtw	3	3	3	4	3	3	3
hdtw	2	4	4	3	5	4	4
kdf	3	3	4	3	3	3	3
hdf	3	3	3	3	3	3	3

A. Clustering Analysis

Let us analyse the clusters found for the two different profile setups.

1) Distinct Profiles: Let us first consider the distinct profile case. Table I provides an overview of the results we obtained. Each row represents one of these variations whilst the first 5 columns show the number of clusters found per run. The following columns show the mode value/s of the number of clusters for the set of runs and the median. In order to keep the following tables and graphs clear and compact we have abbreviated the various experimental cases as follows:

- k : is used when the clustering technique used was K-medoids.
- h : is used for when Hierarchical Clustering was utilised
- eu : stands for the Euclidean distance metric
- dtw : signifies the use of Dynamic Time Warping
- df : indicates the use of the derived features

An example abbreviation is 'keu'. This abbreviation stands for the experimental setup that used K-Medoids as clustering algorithm with the use of the Euclidean distance metric on features directly related to the state. In contrast 'kdf' is the same but with the use of the derived features.

Important to note that in the 'kdf' and the 'hdf' median cases the resulting clustering of the agents corresponded to the perfect distribution of the profiles to the agents. Furthermore the 'kdtw' median case resulted in near perfect clustering; cluster A contained 31 out of 33 athlete agents and one retiree agent, cluster B contained the remaining 2 athlete agents and the rest of the retiree agents with the last cluster only containing all of the Worker type agents. In the case of 'keu' even though three clusters are found, one of the clusters contains most of the agents, with the second cluster containing 3 athlete agents and 7 retiree agents and the last cluster containing only one athelete agent. Similar happened with the 'heu' case, where one cluster contains most agents and the others only have a few.

2) Overlapping Profiles: For the overlapping profiles the results are shown in Table II. The major outcomes that can be taken away from the table are that in hardly any run three clusters were found. Furthermore, it is interesting to note that in the methods using the derived features, one of the clusters contained most, if not all, of the 'newWorker' type agent. Finally, a similar behaviour can be seen in the case of the Hierarchical Clustering using DTW, but instead of dividing

TABLE II
NUMBER OF CLUSTERS RETURNED BY EACH EXPERIMENTAL CASE (FOR THE OVERLAPPING PROFILE CASE)

	run1	run2	run3	run4	run5	Mode	Median
keu	3	5	5	2	2	2,5	3
heu	2	2	2	2	2	2	2
kdtw	2	6	6	2	3	2,6	3
hdtw	2	2	2	2	2	2	2
kdf	2	2	2	2	2	2	2
hdf	2	2	2	6	2	2	2

the 'newWorker' agents from the rest, it divided the 'Athlete' type agents from the 'newWorker' and 'newAthlete' agents.

B. Reinforcement Learning Results

Given the clusters that have been found, we will study the impact on the RL performances now.

1) Distinct Profiles: Within this subsection we will be describing the results found by the reinforcement learning analysis in terms of reward over time for the case of the distinct profiles. The first table presented, Table III, shows a comprehensive overview of all of the experiments run. This overview clearly shows that several of the cluster-based approaches obtain higher cumulative rewards compared to the pooled and separate cases. It seems that the derived features perform best, the Euclidean distance approaches perform worst, and the DTW approaches reside in the middle (while still performing better than the separate and pooled cases). The clustering technique does not seem to have a severe impact on the overall rewards that are obtained.

TABLE III
CUMULATIVE AVERAGE DAILY REWARD FOR ALL EXPERIMENTAL CASES (DISTINCT)

KEU	HEU	KDTW	HDTW
340.1	419.4	740.2	778.7
KDF	**HDF**	**POOLED**	**SEPARATE**
878.3	889.5	310.0	681.9

Let us look into the rewards collected over time. To ease comparison we have selected only four of the six clustering methods used. To make the selection we have excluded the two worst performing methods in terms of cumulative average daily reward (both of the Euclidean distance cases). Note that the performances during the warm up period are identical as a random policy is followed. Analysing Fig. 1 we can notice a recurring pattern that holds for all plots that will be presented, all of the reward trends have the same kind of 'rhythm' to them. This is causes by the fatigue concept. What is important to note in this particular figure is how K-Medoids with DTW has consistently the lowest average reward. This suggests that this method (in this particular case) was the least effective (out of the four selected ones) in aiding the reinforcement learners in producing effective policies.

Figure 2 illustrates the final two selected clustering methods and compares it to the *pooled* and *separate* approaches. To have a comprehensive comparison, we chose the best

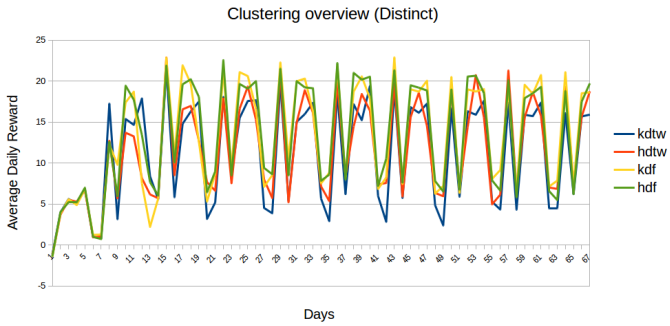

Fig. 1. Plot of the Average Daily Reward over time for the four better performing clustering methods

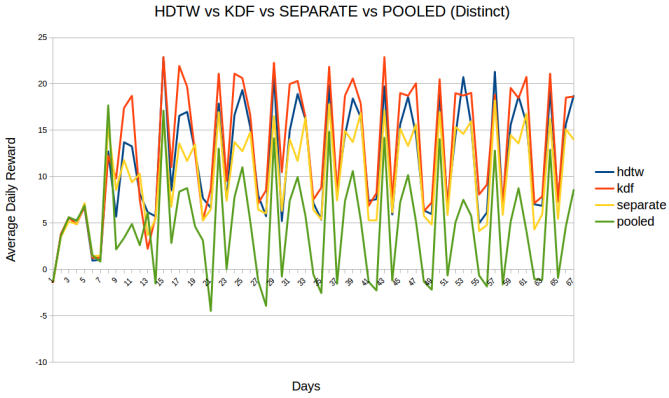

Fig. 2. Plot of the Average Daily Reward over time comparing the two selected clustering methods and the two non-clustering methods (Separate and Pooled)

clustering technique for the non-derived features and the best one for the derived features. In this figure we can easily notice how poorly the *pooled* aided at the creation of a good policy. Furthermore, the daily average reward resulting from *separate* appears to always be below our selected methods.

In order to draw critical conclusions from the comparison we used the Wilcoxon signed-rank test on all possible combinations of the selected methods to find potential statistical differences (as reported in Table IV).

TABLE IV
TABLE OF RETURNED WILCOXON P-VALUES FOR ALL OF THE SELECTED EXPERIMENTAL METHODS (DISTINCT)

hdtw vs kdf	pvalue=3.0675e-06***
hdtw vs separate	pvalue=8.2819e-07***
hdf vs separate	pvalue=3.0421e-10***
separate vs pooled	pvalue=5.1079e-12***
hdtw vs pooled	pvalue=4.8880e-12***
kdf vs pooled	pvalue=5.8275e-12***

The table shows that all of the Fig. 2 plotted lines are indeed statistically different from each-other (with the highest rating).

2) Overlapping Profiles: In this subsection we repeat the analysis in the same mode as previously described, but on the results obtained by the overlapping profiles.

TABLE V
CUMULATIVE AVERAGE DAILY REWARD FOR ALL EXPERIMENTAL CASES (OVERLAPPING)

KEU	HEU	KDTW	HDTW
1327.7	1281.7	1380.5	1248.9
KDF	**HDF**	**POOLED**	**SEPARATE**
1586.9	1662.1	1312.4	1322.5

Similarly to Table III, Table V shows a comprehensive overview of all of the experiments done within the overlapping profiles case by illustrating the cumulative average daily rewards. We see that again the derived features perform best, also better than the pooled and separate approaches. This is positive, since the clustering is less obvious for this case.

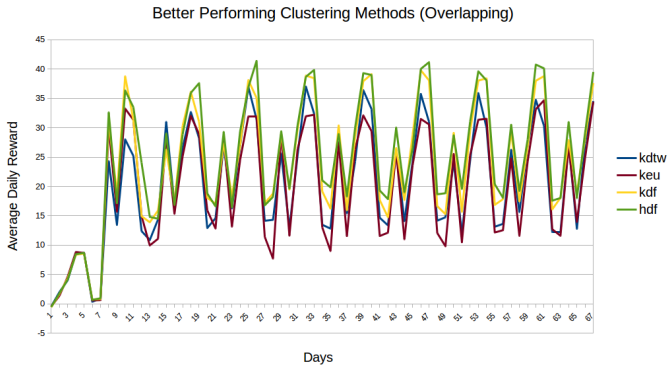

Fig. 3. Plot of the Average Daily Reward over time for the four better performing clustering methods

By selecting the best four clustering methods, with the same criteria as before, we have therefore discarded the two Hierarchical Clustering cases not utilising the derived features. Fig. 3 shows the results. Here we observe that the K-Medoids Euclidean method is consistently scoring the lowest average reward, and close to it is the K-Medoids DTW. This once again illustrates the enhanced difficulty in clustering that the 'overlapping profiles' have, compared to the 'distinct' case.

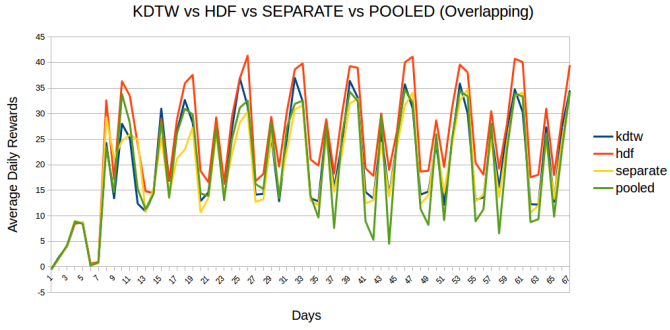

Fig. 4. Plot of the Average Daily Reward over time comparing the two selected clustering methods and the two non-clustering methods (Separate and Pooled)

As of last, Fig. 4 shows the two chosen clustering methods compared to the case of *pooled* and *separate*. For clarity, the selection criterion of the final two clustering methods is the same as the one used in the choice of the final two clustering methods in the 'distinct profile' case. Furthermore, the *pooled* case is once again the lowest of all cases, but is not reaching negative values as it was happening in the other profile case. This, plus the overall rise in average reward across all methods can be attributed by the lack of the 'Retired' agent profile combined with the profile's low maximum fatigue threshold.

TABLE VI
TABLE OF RETURNED WILCOXON P-VALUES FOR ALL OF THE SELECTED EXPERIMENTAL METHODS (OVERLAPPING)

kdtw vs hdf	pvalue=8.6357e-12***
kdtw vs separate	pvalue=0.0005***
hdf vs separate	pvalue=1.0275e-11***
separate vs pooled	pvalue=0.7477
kdtw vs pooled	pvalue=0.0456*
hdf vs pooled	pvalue=2.1030e-12***

Table VI presents the significance results. We want to bring to the attention the now non-statistically significant difference between the *separate* and the *pooled* methods and how our selected DTW method, whilst remaining statistically significant, has now a one-star p-value when compared to the *pooled* method in contrast to the three-star significance when the same comparison was made in the 'distinct profiles' scenario. Nonetheless, even though the 'overlapping profiles' case caused the clustering methods to produce what seemed like worst clusters, we still outperformed both the *separate* and *pooled* case in a statistically significant manner. Therefore showing the benefit of using cluster based reinforcement learning.

VII. DISCUSSION AND FUTURE WORK

With this study we set on exploring in more depth the benefits that cluster-based reinforcement learning can have on personalisation in the health and wellbeing domain. We set up our study in-line with the related work we have found in this field, and expanded the analysis on the different cluster methodologies that can be used in this setting.

Our results show that with distinct profiles the clustering methods utilising DTW and the derived features produced good clusters that were either perfectly matching the profile assignments or extremely close to it. For overlapping profiles, we see that a logical division of the agents was made but it still remains one that does not match the original assignment of the profiles to the agents. For both cases we outperform the *separate* and the *pooled* reinforcement learning approaches. Here, the derived features approach performs best, but the dynamic time warping also performs reasonably well. This seems even more remarkable given the somewhat poor clustering that resulted in the case of the overlapping agent profiles. This finding supports our initial intuition and the findings brought forth by [7], [8].

As future work it would be good to expand on our study and test other types of clustering techniques to see how the

reinforcement learner reacts to potentially different patterns in the clusters. Another interesting variation to study would be to dynamically change the clusters over the course of the simulation, similarly as the reinforcement learner is continuously updating its policy over time. Furthermore, we want to apply the approach in a real life health setting and see whether outcomes improve when using our approach compared to alternative approaches. Finally, as mentioned in the related work, another interesting aspect to investigate would be the use of transfer learning and therefore dropping the assumption that all agents are generated in the simulation at the same point in time.

REFERENCES

[1] H. Fan and M. S. Poole, "What is personalization? perspectives on the design and implementation of personalization in information systems," *Journal of Organizational Computing and Electronic Commerce*, vol. 16, no. 3-4, pp. 179–202, 2006.

[2] E. Vasilyeva, M. Pechenizkiy, and S. Puuronen, "Towards the framework of adaptive user interfaces for ehealth," in *Computer-Based Medical Systems, 2005. Proceedings. 18th IEEE Symposium on*. IEEE, 2005, pp. 139–144.

[3] T. De Pessemier, S. Dooms, and L. Martens, "Context-aware recommendations through context and activity recognition in a mobile environment," *Multimedia Tools and Applications*, vol. 72, no. 3, pp. 2925–2948, 2014.

[4] Y. H. Cho, J. K. Kim, and S. H. Kim, "A personalized recommender system based on web usage mining and decision tree induction," *Expert systems with Applications*, vol. 23, no. 3, pp. 329–342, 2002.

[5] M. Hoogendoorn and B. Funk, *Machine Learning for the Quantified Self: On the Art of Learning from Sensory Data*. Springer, 2017, vol. 35.

[6] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1633–1685, 2009.

[7] F. Zhu, J. Guo, Z. Xu, P. Liao, and J. Huang, "Group-driven reinforcement learning for personalized mhealth intervention," *arXiv preprint arXiv:1708.04001*, 2017.

[8] A. el Hassouni, M. Hoogendoorn, M. van Otterlo, and E. Barbaro, "Personalization of health interventions using cluster-based reinforcement learning," *arXiv preprint arXiv:1804.03592*, 2018.

[9] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*, vol. 10, no. 16. Seattle, WA, 1994, pp. 359–370.

[10] M. Wiering and M. Van Otterlo, "Reinforcement learning," *Adaptation, learning, and optimization*, vol. 12, 2012.

[11] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[12] I. Hochberg, G. Feraru, M. Kozdoba, S. Mannor, M. Tennenholtz, and E. Yom-Tov, "Encouraging physical activity in patients with diabetes through automatic personalized feedback via reinforcement learning improves glycemic control," *Diabetes Care*, vol. 39, no. 4, pp. e59–e60, 2016.

[13] M. G. Lagoudakis and R. Parr, "Least-squares policy iteration," *Journal of machine learning research*, vol. 4, no. Dec, pp. 1107–1149, 2003.

[14] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[15] B. M. Silva, J. J. Rodrigues, I. de la Torre Díez, M. López-Coronado, and K. Saleem, "Mobile-health: A review of current state in 2015," *Journal of biomedical informatics*, vol. 56, pp. 265–272, 2015.

[16] X. Jin and J. Han, "K-medoids clustering," in *Encyclopedia of Machine Learning and Data Mining*. Springer, 2016, pp. 1–3.

[17] M. L. Zepeda-Mendoza and O. Resendis-Antonio, "Hierarchical agglomerative clustering," in *Encyclopedia of Systems Biology*. Springer, 2013, pp. 886–887.