The influence of warm start on the online personalized RL method for the mobile health intervention

Feiyun ZHU

2016 年 8 月 30 日

Contents

Abstract Online personalized reinforcement learning (RL) is more and more popular for the mobile health intervention. It can personalize the type, mode and dose of intervention according to user's ongoing status and needs. However from the perspective of the methodology, the online RL has two drawbacks: 1) it does not have a good initialization of model parameters; 2) at the beginning of the online learning, the RL methods have too few samples to support the actor-critic RL updates, which easily leads to some sub-optimal local minimal that does not perform well. Those two drawbacks both delay the online learning process before achieving good results, which is highly expensive in time costs and risky for the peoples involved to abandon the mobile health intervention.

To address the above two problems, we propose a new online RL methodology that make full use of the data accumulated in the former study as well as the knowledge obtained previously. We hope to highly accelerate the online learning process. Besides, we design an experiment to verify which of the two improvement really makes a difference. Experiment results show dramatically improvement has been obtained, especially at the beginning of the online RL.

Specifically in this draft, we have three mobile health learning schemes:

1. The batch RL. For this learning scheme, we generate the (state, action and immediate reward) trajectory via the micro random trials, where the intervention is given in the probability of 50%. Given all these data, we use the batch actor-cirtc reinforcement learning algorithm to learn the policy. Specifically, the critic update is via the LSTDQ; the actor update is via the fmincon.
2. The 2nd scheme is the personalized online learning, where the RL algorithm interacts with the environment and improves in the process. This method usually needs a warm start. One straight way is to collect a small trajectory of data via micro random trial; and then starts the online learning. However, this method greatly delay the learning process, which is very costly in the sense of time involved in the study. That is, it needs to randomly run the study, to collect a set of data and then to start the learning.
3. The 3rd schemes deal with the problem of the 2nd method. It makes use of the data from the 1st study and treat the learnt parameter from the 1st study as a warm study for the online RL of the new users. This scheme is supposed to learn much faster than the 2nd methods

Notations

1. We use the upper case bold Roman letters to denote matrices and lower case bold Roman letters to represent vectors. For a matrix $\mathbf{Y} = [Y_{ln}] \in \mathbb{R}^{L \times N}$, we denote its l^{th} row and n^{th} column via $\mathbf{y}^l$ and $\mathbf{y}_n$ respectively. We use the uppercase letter to denote the random variable and the lower case to represent the corresponding realization.
2. $S_i \in \mathbb{R}^{Lo}$ is the random state at the i-th decision point, and s_i is the corresponding realization.
3. $f(S_i) \in \mathbb{R}^{Lf}$ is the value feature via the basic functions. In this draft, we have three choices: 1) the state vector itself, 2) the polynomial basic function and 3) the radial basis function (RBF).
4. $\mathbf{x}(f(S_i), A_i) \in \mathbb{R}^{Lh}$ is a vector function to achieve the feature for the linear value approximate. The definition is as follows

   $$\mathbf{x}(f(S_i), A_i) = \left[1, f(S_i)^T, A_i, A_i f(S_i)^T\right]^T \in \mathbb{R}^{Lh=2\times Lf+2}.$$

   In the following, we use $\mathbf{x}(S_i, A_i)$ or $\mathbf{x}_i$ to denote $\mathbf{x}(f(S_i), A_i)$ for simple.
5. $\mathbf{z}(S_i) \in \mathbb{R}^{Lg}$ is a vector function to construct the policy feature. Specifically, we use the state variables directly along with a constant 1, i.e. $\mathbf{z}(S_i) = \left[S_i^T, 1\right]^T \in \mathbb{R}^{Lo+1}$. For the sake of convenience, we usually use $\mathbf{z}_i \in \mathbb{R}^{Lg}$ to represent $\mathbf{z}(S_i) \in \mathbb{R}^{Lg}$.
6. $\pi_\theta(a \mid s) = \dfrac{\exp\left(a\mathbf{z}(s)^T\theta\right)}{1+\exp\left(\mathbf{z}(s)^T\theta\right)}$ is the policy model, where θ is the model parameter and $a \in \{0, 1\}$ is the binary action.
7. N denotes the number of people involved in the study; Ls is the number of entries in the state vector S_i; Lf is the feature length of $f(S_i)$ after basic function; Lh is the feature length of $\mathbf{x}(f(S_i), A_i)$, which is the feature for the linear value approximate; Lg denotes the length of policy feature.

Organization of the rest of the paper The rest of this paper is organized as follows: in Section 1, the batch reinforcement learning methods are introduced. Those method make use of the data generated via the micro-randomed trials. That is, they use the data ready there; they do not need to generate the trajectory in the online learning process. In Section 2, the algorithms for the regular online RL methods, without warm starts, are provided. The algorithms for the online RL methods with warm start are given in Section 3. The details on the simulation data, the performance metric as well as the value features are given in the Section 4.

1 Algorithm for the batch actor-critic RL

1.1 Data Collection via the Micro-Randomized Trials

1. Section 4.1 introduces the details on the data generation (simulation) model (i.e. the MDP model), which is used to generate the MDP trajectories. Specifically, we need to set the following parameters

   (a) N the number of people involved in the data set.

   (b) T the trajectory length;

   (c) σ_r and σ_s are the noise strength when simulating the reward and the state respectively.

   (d) σ_b is the variance ofthe Gaussian noise put in the basic $\boldsymbol{\beta}$ to make each individual's MDP different from each other. The value of σ_b indicates how different the people are.

2. Draw the trajectories for all the N people; each trajectory includes T time points. Specifically at each time point and for each individual in the trajectory, we have a sample tuple, including four elements: the current state, action, reward and next-state as follows

$$\mathcal{D} = \{\mathcal{D}_n\}_{n=1}^{N}, \qquad \text{where } \mathcal{D}_n = \left\{\left(s_i, a_i, r_i, s_i^{'}\right) \mid i = 1, 2, \cdots, T\right\}. \tag{1}$$

   where the action is randomly chosen with probability of 50% to give the intervention. So we totally have $NT = N \times T$ time points drawn from all the N individuals. Collecting all the data, we have the following sample matrices

$$\begin{aligned} \mathbf{X} &= [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_N] \in \mathbb{R}^{L \times NT} \\ \mathbf{Y} &= [\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_N] \in \mathbb{R}^{L \times NT} \\ \mathbf{r} &= \left[\mathbf{r}_1^T, \mathbf{r}_2^T, \cdots, \mathbf{r}_N^T\right]^T \in \mathbb{R}^{NT}, \end{aligned} \tag{2}$$

   where the n-th individual's data is collected in $\{\mathbf{X}_n, \mathbf{Y}_n, \mathbf{r}_n\}$ as follows

$$\begin{aligned} \mathbf{X}_n &= [\mathbf{x}(s_1, a_1), \mathbf{x}(s_2, a_2) \cdots, \mathbf{x}(s_T, a_T)] \in \mathbb{R}^{L \times T} \\ \mathbf{Y}_n &= \left[\sum_a \mathbf{x}\left(s_1^{'}, a\right) \pi_\theta\left(a \mid s_1^{'}\right), \cdots, \sum_a \mathbf{x}\left(s_T^{'}, a\right) \pi_\theta\left(a \mid s_T^{'}\right)\right] \in \mathbb{R}^{L \times T} \\ \mathbf{r}_n &= [r_1, r_2, \cdots, r_T]^T \in \mathbb{R}^T, \end{aligned}$$

   where $\mathbf{X}_n$ collects the value feature for all the current point, $\mathbf{Y}_n$ has all value feature for the next point, and $\mathbf{r}_n$ contains all the immediate reward. For simplicity, we use $\mathbf{x}_n = \mathbf{x}(s_n, a_n)$ to denote the feature vector at the n^{th} time point for the value approximate model here and in the rest of this draft.

1.2 Algorithm for the batch discount reward & contextual bandit via the LSTDQ & Fmincon

1.2.1 Algorithm

1. Generate the trajectories for N people, each with the length of T, and organize them based Section 1.1.
2. **Input parameters** for the discount reward method and the contextual bandit method.
   - (a) ζ_c is the strength of ℓ_2-constraint on $\mathbf{w}$, which is the parameter for the value function.
   - (b) ζ_a is the strength of ℓ_2-constraint on θ, which is the parameter for the policy function.
   - (c) $\gamma \in [0, 1]$ is a known ***discount factor*** (under assumptions does not depend on t). To get the contextual bandit method, we simply set $\gamma = 0$.
3. **Repeat**
   - (a) ***Critic update to get*** *the optimal* $\widehat{\mathbf{w}}_0$ *as follows*

     $$\widehat{\mathbf{w}}_0 = \left(\zeta_c \mathbf{I} + \frac{1}{NT}\sum_{i=1}^{NT} \mathbf{x}_i \left(\mathbf{x}_i - \gamma \mathbf{y}_{i+1}\right)^T\right)^{-1} \left(\frac{1}{NT}\sum_{i=1}^{NT} \mathbf{x}_i r_i\right),$$

     where $\mathbf{x}_i = \mathbf{x}\left(f\left(S_i\right), A_i\right)$ is the feature vector at decision point i for the value funciton; $\mathbf{y}_{i+1} = \sum_a \mathbf{x}\left(f\left(S_{i+1}\right), a\right) \pi_\theta\left(a \mid S_{i+1}\right)$ is the feature at the next time point; r_i is the immediate reward at the i-th point; ζ_c is the balancing parameter for the ℓ_2 constraint to avoid singular solver. Using the notations in Eq. (2), we have a compact critic update solution

     $$\widehat{\mathbf{w}}_0 = \left[\zeta_c \mathbf{I} + \frac{1}{NT}\mathbf{X}\left(\mathbf{X} - \gamma \mathbf{Y}\right)^T\right]^{-1} \left(\frac{1}{NT}\mathbf{X}\mathbf{r}\right).$$

   - (b) ***Actor update*** to get $\hat{\theta}_0 = \arg\max_\theta \hat{J}(\theta, \widehat{\mathbf{w}}_0)$, which is equivalent to

     $$\min_\theta \mathcal{J}\left(\theta\right) = -\frac{1}{NT}\sum_{i=1}^{NT} \sum_{a \in \{0,1\}} Q\left(S_i, a; \widehat{\mathbf{w}}_0\right) \cdot \pi_\theta\left(a|S_i\right) + \frac{\zeta_a}{2}\left\|\theta\right\|_2^2, \qquad (3)$$

     where ζ_a is a balancing parameter for the ℓ_2 constraint on θ to prevent over-fitting; $Q\left(S_i, a; \widehat{\mathbf{w}}_0\right) = \mathbf{x}\left(S_i, a\right)^T \widehat{\mathbf{w}}_0$ is the current estimated value.

   Until convergence

   Output: the final parameter $\hat{\theta}_0$ in the policy function and $\widehat{\mathbf{w}}_0$ in the value approximate.

1.2.2 Appendix: the gradient for (3)

The gradient for (3) is as follows

$$\nabla_\theta \mathcal{J}(\theta) = \frac{1}{NT}\sum_{i=1}^{NT}\sum_{a\in\{0,1\}} Q\left(S_i, a; \widehat{\mathbf{w}}_0\right)\cdot\nabla_\theta\pi_\theta\left(a|S_i\right) + \zeta_a\theta$$

where $\pi_\theta\left(a|S_i\right) = \frac{\exp\left(a\theta^T\mathbf{z}_i\right)}{1+\exp\left(\theta^T\mathbf{z}_i\right)}$, $\mathbf{z}_i = \mathbf{z}\left(S_i\right)$ is the feature for policy, and

$$\begin{aligned}\nabla_\theta\pi_\theta\left(a|S_i\right) &= \frac{\left(\nabla_\theta \exp\left(a\theta^T\mathbf{z}\right)\right)\left(1+\exp\left(\theta^T\mathbf{z}\right)\right) - \left(\nabla_\theta\left(1+\exp\left(\theta^T\mathbf{z}\right)\right)\right)\exp\left(a\theta^T\mathbf{z}\right)}{\left(1+\exp\left(\theta^T\mathbf{z}\right)\right)^2}. \\ &= \frac{\exp\left(a\theta^T\mathbf{z}\right)\left(a\mathbf{z}\right)\left(1+\exp\left(\theta^T\mathbf{z}\right)\right) - \exp\left(\theta^T\mathbf{z}\right)\mathbf{z}\exp\left(a\theta^T\mathbf{z}\right)}{\left(1+\exp\left(\theta^T\mathbf{z}\right)\right)^2} \\ &= \frac{\exp\left(a\theta^T\mathbf{z}\right)}{1+\exp\left(\theta^T\mathbf{z}\right)}\cdot\mathbf{z}\frac{a\left(1+\exp\left(\theta^T\mathbf{z}\right)\right) - \exp\left(\theta^T\mathbf{z}\right)}{1+\exp\left(\theta^T\mathbf{z}\right)} \\ &= \pi_\theta\left(a \mid S_i\right)\mathbf{z}\left(a - \pi_\theta\left(1 \mid S_i\right)\right).\end{aligned}$$

For $a \in \{0,1\}$, we have

$$\begin{aligned}\nabla_\theta\pi_\theta\left(0|\mathbf{z}\right) &= \frac{-\exp\left(\theta^T\mathbf{z}\right)\mathbf{z}}{\left(1+\exp\left(\theta^T\mathbf{z}\right)\right)^2} \\ \nabla_\theta\pi_\theta\left(1|\mathbf{z}\right) &= \frac{\exp\left(\theta^T\mathbf{z}\right)\mathbf{z}}{\left(1+\exp\left(\theta^T\mathbf{z}\right)\right)^2} = -\nabla_\theta\pi_\theta\left(0|\mathbf{z}\right).\end{aligned}$$

1.3 Algorithm for the batch average reward via the LSTDQ & Fmincon

1.3.1 Algorithm

1. Generate the trajectories for N people, each with the trajectory length as T, and organize them based on Section 1.1.
2. **Input parameters** for the discount and contextual bandit algorithm
   (a) ζ_c is the strength of ℓ_2-constraint on **w**, which is the parameter for the value function.
   (b) ζ_a is the strength of ℓ_2-constraint on θ, which is the parameter for the policy function.
3. **Repeat**
   (a) ***Critic update to get*** *the optimal* $\widehat{\mathbf{w}}_0$ *as follows*

$$\widehat{\mathbf{w}}_0 = \left(\zeta_c\mathbf{I} + \frac{1}{NT}\sum_{i=1}^{NT}\left(\mathbf{x}_i - \bar{\mathbf{x}}_0\right)\left(\mathbf{x}_i - \gamma\mathbf{y}_{i+1}\right)^T\right)^{-1}\left(\frac{1}{NT}\sum_{i=1}^{NT}\left(\mathbf{x}_i - \bar{\mathbf{x}}_0\right)r_i\right), \quad (4)$$

where $\mathbf{x}_i = \mathbf{x}\left(f\left(S_i\right), A_i\right)$ is the feature vector at decision point i for the value funciton; $\mathbf{y}_{i+1} = \sum_a \mathbf{x}\left(f\left(S_{i+1}\right), a\right) \pi_\theta\left(a \mid S_{i+1}\right)$ is the feature at the next time point; r_i is the immediate reward at the i-th point; ζ_c is the balancing parameter for the ℓ_2 constraint to avoid singular solver; $\bar{\mathbf{x}}_0 = \frac{1}{NT}\sum_{i=1}^{NT}\mathbf{x}_i$ is the mean of all the current value features.

(b) ***Actor update*** to get $\hat{\theta}_0 = \arg\max_\theta \hat{J}(\theta, \widehat{\mathbf{w}}_0)$, which is equivalent to

$$\begin{aligned}\min_\theta \mathcal{J}_t\left(\theta\right) &= -\frac{1}{NT}\sum_{i=1}^{NT}\left(r_i + V\left(S_{i+1}; \widehat{\mathbf{w}}_0\right) - Q\left(S_i, A_i; \widehat{\mathbf{w}}_0\right)\right) + \frac{\zeta_a}{2}\left\|\theta\right\|_2^2, \qquad (5)\\ &= -\frac{1}{NT}\sum_{i=1}^{NT}\left(r_i + \sum_{a\in\mathcal{A}} Q\left(S_{i+1}, a; \widehat{\mathbf{w}}_0\right)\pi_\theta\left(a|S_{i+1}\right) - Q\left(S_i, A_i; \widehat{\mathbf{w}}_0\right)\right) + \frac{\zeta_a}{2}\left\|\theta\right\|_2^2.\end{aligned}$$

Here the actor update and the critic update are done separately; besides, we substitute the value of $\widehat{\mathbf{w}}_0$ instead of the expression of $\widehat{\mathbf{w}}_0$ into the actor objective function. $Q\left(S_i, A_i; \widehat{\mathbf{w}}_0\right)$ is not related to the parameter θ. Removing the irrelevant terms with θ in (5), we have the new objective function

$$\min_\theta \mathcal{J}\left(\theta\right) = -\frac{1}{NT}\sum_{i=1}^{NT}\sum_{a\in\mathcal{A}} Q\left(S_{i+1}, a; \widehat{\mathbf{w}}_0\right)\cdot\pi_\theta\left(a|S_{i+1}\right) + \frac{\zeta_a}{2}\left\|\theta\right\|_2^2, \qquad (6)$$

where ζ_a is a balancing parameter for the ℓ_2 constraint on θ to prevent over-fitting; $Q\left(S_{i+1}, a; \widehat{\mathbf{w}}_0\right) = \mathbf{x}\left(S_{i+1}, a\right)^T \widehat{\mathbf{w}}_0$ is the current estimated value.

Until convergence

Output: the final parameter $\hat{\theta}_0$ in the policy function and $\widehat{\mathbf{w}}_0$ in the value approximate.

1.3.2 Appendixe#1: prove of the critic update (4)

Proof: According to Susan's draft, the basic equation for the actor critic average reward is

$$0 = \sum_{i=1}^{t}\left(r_i - \eta + V\left(S_{i+1}; \mathbf{w}\right) - Q\left(S_i, A_i; \mathbf{w}\right)\right)\begin{pmatrix}1\\ f\left(S_i\right)\left(1 - A_i\right)\\ f\left(S_i\right)A_i\end{pmatrix} - \zeta_a\mathbf{I}\begin{pmatrix}0\\ \mathbf{w}_0\\ \mathbf{w}_1\end{pmatrix} \qquad (7)$$

where $v = \left(v_0^T, v_1^T\right)^T$ and $\eta \in \mathbb{R}$ are unknown variables; $Q\left(S_i, A_i; \mathbf{w}\right) = \mathbf{w}^T\mathbf{x}_i$ and $V\left(S_{i+1}; \mathbf{w}\right) = \mathbf{w}^T\mathbf{y}_{i+1}$. If we using a more general feature, like $\mathbf{x}_i = \mathbf{x}\left(f\left(S_i\right), A_i\right)$ and $\mathbf{y}_{i+1} = \sum_a \mathbf{x}\left(f\left(S_{i+1}\right), a\right)\pi_\theta\left(a \mid S_{i+1}\right)$, (7) is turned into

$$0 = \sum_{i=1}^{t}\left(r_i - \eta + \mathbf{y}_{i+1}^T\mathbf{w} - \mathbf{x}_i^T\mathbf{w}\right)\begin{pmatrix}1\\ \mathbf{x}_i\end{pmatrix} - \zeta_a\mathbf{I}\begin{pmatrix}0\\ \mathbf{w}\end{pmatrix} \qquad (8)$$

There are two rows in (8). Considering the first row, we have

$$0 = \sum_{i=1}^{t} \left(r_i - \eta + \left(\mathbf{y}_{i+1} - \mathbf{x}_i \right)^T \mathbf{w} \right)$$

and

$$\eta = \frac{1}{t} \sum_{i=1}^{t} \left(r_i + \left(\mathbf{y}_{i+1} - \mathbf{x}_i \right)^T \mathbf{w} \right). \tag{9}$$

Considering the last row in (8), we have

$$\sum_{i=1}^{t} \mathbf{x}_i \left(r_i - \eta + \left(\mathbf{y}_{i+1} - \mathbf{x}_i \right)^T \mathbf{w} \right) - \zeta_a \mathbf{w} = 0$$

and

$$\left(\zeta_a \mathbf{I} + \sum_{i=1}^{t} \mathbf{x}_i \left(\mathbf{x}_i - \mathbf{y}_{i+1} \right)^T \right) \mathbf{w} = \sum_{i=1}^{t} \mathbf{x}_i \left(r_i - \eta \right). \tag{10}$$

Substituting the estimator of η in (9) into (10), we have

$$\begin{aligned}
\left(\zeta_a \mathbf{I} + \sum_{i=1}^{t} \mathbf{x}_i \left(\mathbf{x}_i - \mathbf{y}_{i+1} \right)^T \right) \mathbf{w} &= \sum_{i=1}^{t} \mathbf{x}_i \left(r_i - \eta \right) \\
= \sum_{i=1}^{t} \mathbf{x}_i \left(r_i - \frac{1}{t} \sum_{i=1}^{t} \left(r_i + \left(\mathbf{y}_{i+1} - \mathbf{x}_i \right)^T \mathbf{w} \right) \right) \\
= \sum_{i=1}^{t} \mathbf{x}_i r_i - \left[\sum_{i=1}^{t} \mathbf{x}_i \cdot \frac{1}{t} \sum_{i=1}^{t} \left(r_i + \left(\mathbf{y}_{i+1} - \mathbf{x}_i \right)^T \mathbf{w} \right) \right] \\
= \sum_{i=1}^{t} \mathbf{x}_i r_i - \left(\frac{1}{t} \sum_{i=1}^{t} \mathbf{x}_i \right) \left(\sum_{i=1}^{t} \left(r_i + \left(\mathbf{y}_{i+1} - \mathbf{x}_i \right)^T \mathbf{w} \right) \right).
\end{aligned}$$

Let $\bar{\mathbf{x}}_t = \left(\frac{1}{t} \sum_{i=1}^{t} \mathbf{x}_i \right)$, we have the following derivations

$$\begin{aligned}
\left(\zeta_a \mathbf{I} + \sum_{i=1}^{t} \mathbf{x}_i \left(\mathbf{x}_i - \mathbf{y}_{i+1} \right)^T \right) \mathbf{w} &= \sum_{i=1}^{t} \mathbf{x}_i r_i - \bar{\mathbf{x}}_t \left(\sum_{i=1}^{t} \left(r_i + \left(\mathbf{y}_{i+1} - \mathbf{x}_i \right)^T \mathbf{w} \right) \right) \\
&= \sum_{i=1}^{t} \left(\mathbf{x}_i - \bar{\mathbf{x}}_t \right) r_i + \sum_{i=1}^{t} \bar{\mathbf{x}}_t \left(\mathbf{x}_i - \mathbf{y}_{i+1} \right)^T \mathbf{w},
\end{aligned}$$

and

$$\left(\zeta_a \mathbf{I} + \sum_{i=1}^{t} \left(\mathbf{x}_i - \bar{\mathbf{x}}_t \right) \left(\mathbf{x}_i - \mathbf{y}_{i+1} \right)^T \right) \mathbf{w} = \sum_{i=1}^{t} \left(\mathbf{x}_i - \bar{\mathbf{x}}_t \right) r_i.$$

■

1.3.3 Appendix#2: the gradient for (3)

Please check Section 1.2.2 for detail.

2 Algorithm for the online actor-critic RL without Warm-Start

2.1 Online actor-cirtc Contextual bandit

1. **Input**:
   (a) T_{max} the maximum trajectory length in the online learning setting.
   (b) ζ_c is the strength of ℓ_2-constraint on v, which is the parameter for the value function.
   (c) ζ_a the strength of ℓ_2-constraint on θ, which is the parameter for the policy function.
   (d) $\mathbf{B}_0 = \zeta_c \mathbf{I}_{Lh\times Lh} \in \mathbb{R}^{Lh\times Lh}$ and $\mathbf{a}_0 = \mathbf{0} \in \mathbb{R}^{Lh}$.
2. **While** ($t < T_{\text{max}}$)
   (a) Observe context s_t.
   (b) Draw an action a_t according to the probability distribution $\pi_\theta(a|s_t)$.
   (c) Observe an immediate reward r_t;
   (d) ***Critic update to get*** *the optimal* $\widehat{\mathbf{w}}_t$ *at point t via*

$$\widehat{\mathbf{w}}_t = \mathbf{B}_t^{-1}\mathbf{a}_t, \tag{11}$$

   where $\mathbf{B}_t = \mathbf{B}_{t-1} + \mathbf{x}\left(s_t, a_t\right)\mathbf{x}\left(s_t, a_t\right)^T$ and $\mathbf{a}_t = \mathbf{a}_{t-1} + \mathbf{x}\left(s_t, a_t\right) r_t$. Since $\{\mathbf{B}_t\}_{t=1}^T$ is generally invertible, we may use the Sherman–Morrison formula to highly reduce the computational cost of (11), resulting in the following updates

$$\widehat{\mathbf{w}}_t = \widehat{\mathbf{w}}_{t-1} + \mathbf{B}_{t-1}^{-1}\mathbf{x}\left(s_t, a_t\right)\left\{\frac{\mathbf{x}\left(s_t, a_t\right)^T\mathbf{B}_{t-1}^{-1} - r_t}{1 + \mathbf{x}\left(s_t, a_t\right)^T\mathbf{B}_{t-1}^{-1}\mathbf{x}\left(s_t, a_t\right)}\right\}$$

   (e) ***Actor update*** to get $\hat{\theta}_t = \arg\max_{\theta} \hat{J}_t(\theta, \widehat{\mathbf{w}}_t)$, which is equivalent to

$$\min_{\theta} \mathcal{J}_t\left(\theta\right) = -\frac{1}{t}\sum_{i=1}^{t}\sum_{a} Q\left(S_i, a; \mathbf{w}_t\right)\pi_\theta\left(a|S_i\right) + \frac{\zeta_a}{2}\left\|\theta\right\|_2^2,$$

   where ζ_a is a balancing parameter for the ℓ_2 constraint on θ to prevent over-fitting; $Q\left(S_i, a; \widehat{\mathbf{w}}_t\right) = \mathbf{x}\left(S_i, a\right)^T \widehat{\mathbf{w}}_t$ is the current estimated value.
3. **End While**
4. **Output:** the final parameter $\hat{\theta}_t$ in the policy function and $\widehat{\mathbf{w}}_t$ in the value approximate.

2.2 Online actor-cirtc discount reward

1. **Input**:
   (a) T_{max} the maximum trajectory length in the online learning setting.
   (b) ζ_c is the strength of ℓ_2-constraint on v, which is the parameter for the value function.
   (c) ζ_a the strength of ℓ_2-constraint on θ, which is the parameter for the policy function.
   (d) $\gamma \in [0,1]$ is a known ***discount factor*** (under assumptions does not depend on t). To get the contextual bandit method, we simply set $\gamma = 0$.
2. **While** ($t < T_{\text{max}}$)
   (a) Observe context s_t.
   (b) Draw an action a_t according to the probability distribution $\pi_\theta(a|s_t)$.
   (c) Observe an immediate reward r_t;
   (d) ***Critic update to get*** *the optimal* $\widehat{\mathbf{w}}_t$ *at point t via*

$$\widehat{\mathbf{w}}_t = \left(\zeta_c \mathbf{I} + \frac{1}{t}\sum_{i=1}^{t} \mathbf{x}_i \left(\mathbf{x}_i - \gamma \mathbf{y}_{i+1}\right)^T\right)^{-1} \left(\frac{1}{t}\sum_{i=1}^{t} \mathbf{x}_i r_i\right),$$

   where $\mathbf{x}_i = \mathbf{x}\left(f\left(S_i\right), A_i\right)$ is the feature vector at decision point i for the value funciton; $\mathbf{y}_{i+1} = \sum_a \mathbf{x}\left(f\left(S_{i+1}\right), a\right) \pi_{\theta_t}\left(a \mid S_{i+1}\right)$ is the feature at the next time point; r_i is the immediate reward at the i-th point; ζ_c is the balancing parameter for the ℓ_2 constraint to avoid singular solver. Note that after each actor update, we need to re-calculate the next value feature $\mathbf{y}_{i+1} = \sum_a \mathbf{x}\left(f\left(S_{i+1}\right), a\right) \pi_{\theta_t}\left(a \mid S_{i+1}\right)$ based on the new obtained policy parameter θ_t.
   (e) ***Actor update*** to get $\hat{\theta}_t = \arg\max_\theta \hat{J}_t(\theta, \widehat{\mathbf{w}}_t)$, which is equivalent to

$$\min_\theta \mathcal{J}_t\left(\theta\right) = -\frac{1}{t}\sum_{i=1}^{t}\sum_a Q\left(S_i, a; \mathbf{w}_t\right) \pi_\theta\left(a|S_i\right) + \frac{\zeta_a}{2}\left\|\theta\right\|_2^2,$$

   where ζ_a is a balancing parameter for the ℓ_2 constraint on θ to prevent over-fitting; $Q\left(S_i, a; \widehat{\mathbf{w}}_t\right) = \mathbf{x}\left(S_i, a\right)^T \widehat{\mathbf{w}}_t$ is the current estimated value.
3. **End While**
4. **Output:** the final parameter $\hat{\theta}_T$ in the policy function and $\widehat{\mathbf{w}}_t$ in the value approximate.

2.3 Online actor-cirtc Average Reward

1. **Input**:
   (a) $T_{\max}$ the maximum trajectory length in the online learning setting.
   (b) ζ_c is the strength of ℓ_2-constraint on v, which is the parameter for the value function.
   (c) ζ_a is the strength of ℓ_2-constraint on θ, which is the parameter for the policy function.
2. **While** ($t < T_{\max}$)
   (a) Observe context s_t.
   (b) Draw an action a_t according to the probability distribution $\pi_\theta(a_t|s_t)$.
   (c) Observe an immediate reward r_t.
   (d) ***Critic update to get*** *the optimal* $\widehat{\mathbf{w}}_t$ *at point t via*

   $$\widehat{\mathbf{w}}_t = \left(\zeta_c \mathbf{I} + \frac{1}{t}\sum_{i=1}^{t}\left(\mathbf{x}_i - \bar{\mathbf{x}}_0\right)\left(\mathbf{x}_i - \gamma \mathbf{y}_{i+1}\right)^T\right)^{-1}\left(\sum_{i=1}^{t}\left(\mathbf{x}_i - \bar{\mathbf{x}}_0\right) r_i\right),$$

   where $\mathbf{x}_i = \mathbf{x}\left(f\left(S_i\right), A_i\right)$ is the feature vector at decision point i for the value funciton; $\mathbf{y}_{i+1} = \sum_a \mathbf{x}\left(f\left(S_{i+1}\right), a\right)\pi_\theta\left(a \mid S_{i+1}\right)$ is the feature at the next time point; r_i is the immediate reward at the i-th point; ζ_c is the balancing parameter for the ℓ_2 constraint to avoid singular solver. $\bar{\mathbf{x}}_0 = \frac{1}{NT}\sum_{i=1}^{NT}\mathbf{x}_i$ is the mean of all the current value features. Note that after each actor update, we need to re-calculate the next value feature $\mathbf{y}_{i+1} = \sum_a \mathbf{x}\left(f\left(S_{i+1}\right), a\right)\pi_{\theta_t}\left(a \mid S_{i+1}\right)$ based on the new obtained policy parameter θ_t.

   (e) ***Actor update*** to get $\hat{\theta}_t = \arg\max_\theta \hat{J}_t(\theta, \widehat{\mathbf{w}}_t)$, which is equivalent to

   $$\min_\theta \mathcal{J}_t\left(\theta\right) = -\frac{1}{t}\sum_{i=1}^{t}\left(r_i + \sum_{a\in\mathcal{A}} Q\left(S_{i+1}, a; \widehat{\mathbf{w}}_t\right)\pi_\theta\left(a|S_{i+1}\right) - Q\left(S_i, A_i; \widehat{\mathbf{w}}_t\right)\right) + \frac{\zeta_a}{2}\left\|\theta\right\|_2^2.$$

   Here the actor update and the critic update are done separately; besides, we substitute the value of $\widehat{\mathbf{w}}_t$ instead of the expression of $\widehat{\mathbf{w}}_t$ into the actor objective function. $Q\left(S_i, A_i; \widehat{\mathbf{w}}_t\right)$ is not related to the parameter θ. Removing the irrelevant terms with θ in (5), we have the new objective function

   $$\min_\theta \mathcal{J}_t\left(\theta\right) = -\frac{1}{t}\sum_{i=1}^{t}\sum_{a\in\mathcal{A}} Q\left(S_{i+1}, a; \widehat{\mathbf{w}}_t\right)\cdot\pi_\theta\left(a|S_i\right) + \frac{\zeta_a}{2}\left\|\theta\right\|_2^2,$$

   where ζ_a is a balancing parameter for the ℓ_2 constraint on θ to prevent over-fitting; $Q\left(S_i, a; \widehat{\mathbf{w}}_t\right) = \mathbf{x}\left(S_i, a\right)^T \widehat{\mathbf{w}}_t$ is the current estimated value.
3. **End While**
4. **Output:** the final parameter $\hat{\theta}_t$ in the policy function and $\widehat{\mathbf{w}}_t$ in the value approximate.

3 Algorithms for the online actor-cirtc RL with Warm-Start

There are two kinds of differences between the RL methods with warm start and those RL methods without warm-starts:

1. the RL with warm start make use of the data from the 1st set of βs (i.e. individuals or MDPs), while the RL without warm start does not make any use of the data from 1st set of βs.
2. the RL with warm start takes the learnt value parameter and policy parameter, via the batch RL methods on the 1st set of of data, as the initialization of the parameters in the online learning.

In Sections 3.1, 3.2 and 3.3, we will give the algorithms for the RL method with warm start. Note that the terms in red uses the data generated via the micro-randomed trial based on the 1st set of individuals.

Notations to differentiate the data from 1st set of individual and the the data collected in the online process:

- the varibles with bars are the data from the 1st set of beta, i.e. $\{(\bar{\mathbf{x}}_n, \bar{\mathbf{y}}_n, \bar{r}_n)\}_{n=1}^{NT}$, where N is the number of people involved; T is the trajectory length.
- the normal varibles are the data generated in the online learning, i.e. $\{(\mathbf{x}_i, \mathbf{y}_i, r_i)\}_{i=1}^{t}$, where t is the current trajectory length in the online learning.

3.1 Online actor-cirtc Contextual bandit with warm start

1. **Input**:

   (a) $T_{\max}$ the maximum trajectory length in the online learning setting.

   (b) ζ_c is the strength of ℓ_2-constraint on v, which is the parameter for the value function.

   (c) ζ_a the strength of ℓ_2-constraint on θ, which is the parameter for the policy function.

   (d) the learnt value parameter $\widehat{\mathbf{w}}_0$ and the learnt policy parameter $\hat{\theta}_0$, via the batch RL methods on the data from the 1st set of individuals, as an initialization of the corresponding parameters on the actor-critic RL methods.

2. **While** ($t < T_{\max}$)

   (a) Observe context s_t.

   (b) Draw an action a_t according to the probability distribution $\pi_\theta(a|s_t)$.

   (c) Observe an immediate reward r_t;

(d) ***Critic update to get*** *the optimal $\widehat{\mathbf{w}}_t$ at point t via*

$$\widehat{\mathbf{w}}_t = \left\{\zeta_c \mathbf{I} + \frac{1}{t+1}\left[\frac{1}{NT}\sum_{j=1}^{NT}\bar{\mathbf{x}}_j\bar{\mathbf{x}}_j^T + \sum_{i=1}^{t}\mathbf{x}_i\mathbf{x}_i^T\right]\right\}^{-1}\left[\frac{1}{t+1}\left(\frac{1}{NT}\sum_{j=1}^{NT}\bar{\mathbf{x}}_j\bar{r}_j + \sum_{i=1}^{t}\mathbf{x}_i r_i\right)\right].$$

where $\{\bar{\mathbf{x}}_j\}_{j=1}^{NT}$ is the data from the 1st set of individuals (i.e. N individuals, each is with a trajectory of T points); $(\mathbf{x}_i)_{i=1}^{t}$ is the data that is collected when the RL method interacts with the MDP system; $\mathbf{x}_i = \mathbf{x}\left(f\left(S_i\right), A_i\right)$ is the feature vector at decision point i for the value funciton; $\mathbf{y}_{i+1} = \sum_a \mathbf{x}\left(f\left(S_{i+1}\right), a\right)\pi_{\theta_t}\left(a \mid S_{i+1}\right)$ is the feature at the next time point; r_i is the immediate reward at the i-th point; ζ_c is the balancing parameter for the ℓ_2 constraint to avoid singular solver.

(e) ***Actor update*** to get $\hat{\theta}_t = \arg\max_{\theta} \hat{J}_t(\theta, \widehat{\mathbf{w}}_t)$, which is equivalent to

$$\min_{\theta} \mathcal{J}\left(\theta\right) = -\frac{1}{t+1}\left\{\frac{1}{NT}\sum_{j=1}^{NT}\sum_{a\in\{0,1\}} Q\left(\bar{S}_j, a; \widehat{\mathbf{w}}_t\right)\pi_\theta\left(a|\bar{S}_j\right) + \sum_{i=1}^{t}\sum_{a\in\{0,1\}} Q\left(S_i, a; \widehat{\mathbf{w}}_t\right)\cdot\pi\left(a|S_i\right)\right\} + \frac{\zeta_a}{2}\left\|\theta\right\|_2^2$$

where ζ_a is a balancing parameter for the ℓ_2 constraint on θ to prevent over-fitting; $Q\left(S_i, a; \widehat{\mathbf{w}}_t\right) = \mathbf{x}\left(S_i, a\right)^T\widehat{\mathbf{w}}_t$ is the current estimated value.

3. **End While**

4. **Output:** the final parameter $\hat{\theta}_T$ in the policy function and $\mathbf{w}_t$ in the value approximate.

3.2 Online actor-cirtc discount reward with warm start

1. **Input**:

(a) $T_{\max}$ the maximum trajectory length in the online learning setting.

(b) ζ_c is the strength of ℓ_2-constraint on v, which is the parameter for the value function.

(c) ζ_a the strength of ℓ_2-constraint on θ, which is the parameter for the policy function.

(d) $\gamma \in [0,1]$ is a known ***discount factor*** (under assumptions does not depend on t). To get the contextual bandit method, we simply set $\gamma = 0$.

(e) the learnt value parameter $\widehat{\mathbf{w}}_0$ and the learnt policy parameter $\hat{\theta}_0$, via the batch RL methods on the data from the 1st set of individuals, as an initialization of the corresponding parameters on the actor-critic RL methods.

2. **While** ($t < T_{\max}$)

    (a) Observe context s_t.

    (b) Draw an action a_t according to the probability distribution $\pi_\theta(a_t|s_t)$.

    (c) Observe an immediate reward r_t.

    (d) ***Critic update to get*** *the optimal $\widehat{\mathbf{w}}_t$ at point t via*

    $$\widehat{\mathbf{w}}_t = \left\{\zeta_c\mathbf{I} + \frac{1}{t+1}\left[\frac{1}{NT}\sum_{j=1}^{NT}\boxed{\mathbf{x}}_j\left(\boxed{\mathbf{x}}_j - \gamma\boxed{\mathbf{y}}_{i+1}\right)^T + \sum_{i=1}^{t}\mathbf{x}_i\left(\mathbf{x}_i - \gamma\mathbf{y}_{i+1}\right)^T\right]\right\}^{-1}$$
    $$\left[\frac{1}{t+1}\left(\frac{1}{NT}\sum_{j=1}^{NT}\bar{\mathbf{x}}_j\bar{r}_j + \sum_{i=1}^{t}\mathbf{x}_i r_i\right)\right]$$

    where $\{\bar{\mathbf{x}}_j\}_{j=1}^{NT}$ is the data from the 1st set of individuals (i.e. N individuals, each is with a trajectory of T points); $(\mathbf{x}_i)_{i=1}^{t}$ is the data that is collected when the RL method interacts with the MDP system; $\mathbf{x}_i = \mathbf{x}\left(f\left(S_i\right), A_i\right)$ is the feature vector at decision point i for the value funciton; $\mathbf{y}_{i+1} = \sum_a \mathbf{x}\left(f\left(S_{i+1}\right), a\right)\pi_{\theta_t}\left(a \mid S_{i+1}\right)$ is the feature at the next time point; r_i is the immediate reward at the i-th point; ζ_c is the balancing parameter for the ℓ_2 constraint to avoid singular solver.

    (e) ***Actor update*** to get $\hat{\theta}_t = \arg\max_\theta \hat{J}_t(\theta, \widehat{\mathbf{w}}_t)$, which is equivalent to

    $$\min_\theta \mathcal{J}(\theta) = -\frac{1}{t+1}\left\{\frac{1}{NT}\sum_{j=1}^{NT}\sum_{a\in\{0,1\}} Q\left(\bar{S}_j, a; \widehat{\mathbf{w}}_t\right)\pi_\theta\left(a|\bar{S}_j\right).\right.$$
    $$\left.+\sum_{i=1}^{t}\sum_{a\in\{0,1\}} Q\left(S_i, a; \widehat{\mathbf{w}}_t\right)\cdot\pi\left(a|S_i\right)\right\} + \frac{\zeta_a}{2}\|\theta\|_2^2$$

    where ζ_a is a balancing parameter for the ℓ_2 constraint on θ to prevent over-fitting; $Q\left(S_i, a; \widehat{\mathbf{w}}_t\right) = \mathbf{x}\left(S_i, a\right)^T \widehat{\mathbf{w}}_t$ is the current estimated value.

3. **End While**

4. **Output:** the final parameter $\hat{\theta}_t$ in the policy function and $\mathbf{w}_t$ in the value approximate.

3.3 Online actor-cirtc Average Reward with Warm Start

1. **Input**:

    (a) $T_{\max}$ the maximum trajectory length in the online learning setting.

    (b) ζ_c is the strength of ℓ_2-constraint on v, which is the parameter for the value function.

(c) ζ_a the strength of ℓ_2-constraint on θ, which is the parameter for the policy function.

(d) the learnt value parameter $\widehat{\mathbf{w}}_0$ and the learnt policy parameter $\hat{\theta}_0$, via the batch RL methods on the data from the 1st set of individuals, as an initialization of the corresponding parameters on the actor-critic RL methods.

2. **While** ($t < T_{\max}$)

(a) Observe context s_t.

(b) Draw an action a_t according to the probability distribution $\pi_\theta(a|s_t)$.

(c) Observe an immediate reward r_t;

(d) ***Critic update to get*** *the optimal* $\widehat{\mathbf{w}}_t$ *at point* t *via*

$$\widehat{\mathbf{w}}_t = \left\{\zeta_c \mathbf{I} + \frac{1}{t+1}\left[\frac{1}{NT}\sum_{j=1}^{NT}\left(\boxed{\mathbf{x}}_j - \bar{\mathbf{x}}_0\right)\left(\boxed{\mathbf{x}}_j - \gamma \boxed{\mathbf{y}}_{i+1}\right)^T + \sum_{i=1}^{t}\left(\mathbf{x}_i - \mathbf{x}_0\right)\left(\mathbf{x}_i - \gamma \mathbf{y}_{i+1}\right)^T\right]\right\}^{-1},$$
$$\left\{\frac{1}{t+1}\left[\frac{1}{NT}\sum_{j=1}^{NT}\left(\boxed{\mathbf{x}}_j - \bar{\mathbf{x}}_0\right) r_i + \sum_{i=1}^{t}\left(\mathbf{x}_i - \mathbf{x}_0\right) r_i\right]\right\}$$

where $\mathbf{x}_i = \mathbf{x}\left(f\left(S_i\right), A_i\right)$ is the feature vector at decision point i for the value funciton; $\mathbf{y}_{i+1} = \sum_a \mathbf{x}\left(f\left(S_{i+1}\right), a\right)\pi_\theta\left(a \mid S_{i+1}\right)$ is the feature at the next time point; r_i is the immediate reward at the i-th point; ζ_c is the balancing parameter for the ℓ_2 constraint to avoid singular solver. $\mathbf{x}_0 = \frac{1}{t}\sum_{i=1}^{t}\mathbf{x}_i$ and $\bar{\mathbf{x}}_0 = \frac{1}{NT}\sum_{i=1}^{NT}\bar{\mathbf{x}}_i$ are the means of all the current value features. Note that after each actor update, we need to re-calculate the next value feature $\mathbf{y}_{i+1} = \sum_a \mathbf{x}\left(f\left(S_{i+1}\right), a\right)\pi_{\theta_t}\left(a \mid S_{i+1}\right)$ based on the new obtained policy parameter θ_t.

(e) ***Actor update*** to get $\hat{\theta}_t = \arg\max_{\theta} \hat{J}_t(\theta, \widehat{\mathbf{w}}_t)$, which is equivalent to

$$\min_{\theta} \mathcal{J}(\theta) = -\frac{1}{t+1}\left\{\frac{1}{NT}\sum_{j=1}^{NT}\sum_{a\in\{0,1\}} Q\left(\bar{S}_{j+1}, a; \widehat{\mathbf{w}}_t\right)\pi_\theta\left(a|\bar{S}_{j+1}\right)\right.$$
$$\left. + \sum_{i=1}^{t}\sum_{a\in\{0,1\}} Q\left(S_{i+1}, a; \widehat{\mathbf{w}}_t\right)\cdot \pi_\theta\left(a|S_{i+1}\right)\right\} + \frac{\zeta_a}{2}\|\theta\|_2^2$$

where ζ_a is a balancing parameter for the ℓ_2 constraint on θ to prevent over-fitting; $Q\left(S_i, a; \widehat{\mathbf{w}}_t\right) = \mathbf{x}\left(S_i, a\right)^T \widehat{\mathbf{w}}_t$ is the current estimated value.

3. **End While**

4. **Output:** the final parameter $\hat{\theta}_t$ in the policy function and $\mathbf{w}_t$ in the value approximate.

4 The experiment settings

4.1 Datasets

In the experiments, a trajectory of T (the trajectory length) tuples for each individual is defined as follows

$$\mathcal{D}_T = \left\{ \left(O_0, A_0, R_0 \right), \left(O_1, A_1, R_1 \right), \cdots, \left(O_T, A_T, R_T \right) \right\}, \tag{12}$$

where the initial states and action are generated by $O_0 \sim \text{Normal}_{p_1} \left\{ 0, \Sigma \right\}$ and $A_0 = 0$. Here we have $\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & I_{p-3} \end{bmatrix}$ and $\Sigma_1 = \begin{bmatrix} 1 & 0.3 & -0.3 \\ 0.3 & 1 & -0.3 \\ -0.3 & -0.3 & 1 \end{bmatrix}$. For $t \geq 1$, we have the state generation model as follows

$$\begin{aligned} O_{t,1} &= \beta_1 O_{t-1,1} + \xi_{t,1}, \qquad \text{weather-high is good} \\ O_{t,2} &= \beta_2 O_{t-1,2} + \beta_3 A_{t-1} + \xi_{t,2}, \qquad \text{engagement} \\ O_{t,3} &= \beta_4 O_{t-1,3} + \beta_5 O_{t-1,3} A_{t-1} + \beta_6 A_{t-1} + \xi_{t,3}, \qquad \text{treatment fatigue} \\ O_{t,j} &= \beta_7 O_{t-1,j} + \xi_{t,j}. \quad j = 4, \ldots, p \end{aligned} \tag{13}$$

The corresponding immediate reward model is as follows

$$R_{t+1} = \beta_{14} \times \left[\beta_8 + A_t \times \left(\beta_9 + \beta_{10} O_{t,1} + \beta_{11} O_{t,2} \right) + \beta_{12} O_{t,1} - \beta_{13} O_{t,3} + \varrho_{t,} \right], \tag{14}$$

where $-\beta_{13} O_{t,3}$ is the treatment fatigue. In the above generation model, the $\{\xi_{t,i}\}_{i=1}^{p}$ at the t-th time point are the noise in the state transition model (13) that is drawn from $\{\xi_{t,i}\}_{i=1}^{p} \sim \text{Normal}\left(0, \sigma_s^2\right)$. ϱ_t is the noise in the reward model (14), which is defined as $\varrho_t \sim \text{Normal}\left(0, \sigma_r^2\right)$.

To generate a group of N people that is slightly different from each other, we need N slightly different MDPs. Formally, each MDP is specified by the value of $\boldsymbol{\beta}$, cf. Eqn. (13) and (14). The $\boldsymbol{\beta}s$ is generated the following two steps:

1. we set an initial (i.e. basic)
   $\boldsymbol{\beta}_{\text{basic}} = [0.40, 0.25, 0.35, 0.65, 0.10, 0.50, 0.22, 2.00, 0.15, 0.20, 0.32, 0.10, 0.45, 5]$.
2. To make a group of N people that is slight different, we set the $\{\boldsymbol{\beta}_i\}_{i=1}^{N}$ as

   $$\boldsymbol{\beta}_i = \boldsymbol{\beta}_{\text{basic}} + \boldsymbol{\delta}_i, \qquad \text{for } i \in \{1, 2, \cdots, N\},$$

   where $\boldsymbol{\delta}_i \sim \text{Normal}\left(0, \sigma_b \mathbf{I}_{14}\right)$ and $\mathbf{I}_{14} \in \mathbb{R}^{14\times14}$ is an identity matrix.

To generate the MDP for a future user, we will use this kind of method to generate a new $\boldsymbol{\beta}$. In this way, the $\boldsymbol{\beta}$ for each individual is different from each other; σ_b controls how different the MDPs are, i.e. how different the individuals are.

4.2 Evaluation Metrics for Quantitative Performance

We want to see if the personalized RL methods could learn faster when we initialize online algorithms with $\mathbf{B}_0$, $\mathbf{a}_0$ from the 1st data set. We want to compare them with not using $\mathbf{B}_0$, $\mathbf{a}_0$ from the 1st data set, for different $t < T_{\text{max}}$. We will do these for a variety of t not just T_{max}.

We use one metric to measure the quality of the learnt policy, i.e. the expectation of the long run average reward (i.e. ElrAR) $\mathbb{E}_\pi \left[\eta_{\hat{\theta}}\right]$, where $\hat{\theta}$ is the learnt policy parameter. Intuitively, ElrAR measures how much average reward in the long run we could get by using the learnt policy $\hat{\theta}$. A larger ElrAR corresponds to a better performance. Formally, there are three steps to get the ElrAR:

- For each individual, a very long trajectory of $T = 5,000$ is generated by the using the policy parameter $\hat{\theta}$;
- By averaging the rewards for the last $4,000$ elements from the trajectory, we could get the long run average reward

  $$\eta_{\hat{\theta}} = \frac{1}{T-i} \sum_{j=i}^{T} R_{j+1}\left(S_j, A_j\right)$$

  where $i = 1000, T = 5,000$.
- The expectation of the long run average reward is obtained by averaging $\eta_{\hat{\theta}}$ among all the N individual results the as

  $$\mathbb{E}_\pi \left[\eta_{\hat{\theta}}\right] \approx \frac{1}{N} \sum_{n=1}^{N} \eta_{\hat{\theta}}^{(n)},$$

  where $\eta_{\hat{\theta}}^{(n)}$ is the estimated long run average reward for the n^{th} individual (i.e. MDP).

4.3 Feature construction

In this section, we introduce how to construct the features for the policy function and the value approximate function respectively. The policy feature is easy, which is simply adding one constant into the state vector

$$\mathbf{z}\left(O_i\right) = \left[1, O_t\right]^T \in \mathbb{R}^{Lo+1}.$$

The feature for the value approximation is more complex. It employs the basic function

$$\mathbf{z}\left(O_i, A_i\right) = \left[\left(1 - A_i\right) f\left(O_i\right)^T, A_i f\left(O_i\right)^T\right]^T,$$

where $f\left(O_i\right)$ is a basic function of state O_i. In the follwoing, we will introduce two popular basic function for the reinforcement learning [1, 2].

4.3.1 The polynomial basis [1]

1. Given d state variables $\mathbf{x} = [x_1, x_2, \cdots, x_d] \in \mathbb{R}^{1\times d}$, the ***simplest linear scheme*** uses ***each variable directly*** as a basis function along with a constant function, setting $\phi_0(\mathbf{x}) = 1$ and $\phi_i(\mathbf{x}),\ 0 \le i \le d$. However, most interesting value functions are too complex to be represented this way.
2. This scheme was therefore generalized to the polynomial basis

$$\phi_i(\mathbf{x}) = \prod_{j=1}^{d} x_j^{c_{i,j}} = x_1^{c_{i,1}} x_2^{c_{i,2}}, \cdots, x_d^{c_{i,d}} \in \mathbb{R}^{(n+1)^d},$$

   where each $c_{i,j}$ is an integer between 0 and n. We describe such a basis as an order n polynomial basis. For example, a 2nd order polynomial basis defined over two state variables x and y would have feature vector:

$$\Phi = \left[1, x, y, xy, x^2, y^2, x^2y, xy^2, x^2y^2\right].$$

4.3.2 Radial Basis Function [1]

1. RBF definition: another common common scheme for state vector $\mathbf{x} \in \mathbb{R}^d$; the basic function is a Gaussian:

$$\phi_i(\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\|\mathbf{c}_i - \mathbf{x}\|_2^2}{2\sigma^2}\right),$$

   or a given collection of centers $\mathbf{c}_i$ and variance σ^2.

   (a) the centers c_i are typically evenly along each dimension, leading to n^d centers for d state variables and a given order n.

   (b) σ^2 can be varied but is often set to $\frac{2}{n-1}$

2. RBF's characteristics

   (a) RBFs only generalize locally—changes in one area of the state space do not affect the entire state space.

   (b) Thus, they are suitable for representing value functions that might have discontinuities.

   (c) However, this limited generalization is often reflected in slow initial performance. key

References

[1] G. Konidaris, S. Osentoski, and P. S. Thomas, "Value function approximation in reinforcement learning using the fourier basis," in *Conference on Artificial Intelligence (AAAI)*, 2011.

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2012.