The three actor-critic RL methods, the simulation model and the testing

Feiyun ZHU

2016 年 7 月 30 日

Contents

The 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 as $\mathbf{y}^l$ and $\mathbf{y}_n$ respectively. We use the uppercase letter to denote the random variable.
2. $O_i \in \mathbb{R}^{Lo}$ is the random state at the i-th decision point, and o_i is the corresponding realization. Sometimes, we use S_i and s_i to denote the states as well.
3. $f(O_i) \in \mathbb{R}^{Lf}$ is the value feature via the basic function. In this draft, we have three choices: 1) the state itself, 2) the polynomial basic function and 3) the radial basis function (RBF) [1, 2].
4. $\mathbf{x}(f(O_i), A_i) \in \mathbb{R}^{Lh}$ is a vector function to achieve the feature for the linear value approximate. The definition is as follows

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

   In the following, we use $\mathbf{x}(O_i, A_i)$ or $\mathbf{x}_i$ to denote $\mathbf{x}(f(O_i), A_i)$ for simple.
5. $\mathbf{z}(O_i) \in \mathbb{R}^{Lg}$ is a vector function to construct the policy feature. Specifically, we use the state variables directly along with a constant 1, i.e. $\mathbf{z}(O_i) = \left[O_i^T, 1\right]^T \in \mathbb{R}^{Lo+1}$. For the sake of convenience, we sometimes use $\mathbf{z}_i \in \mathbb{R}^{Lg}$ to represent $\mathbf{z}(O_i) \in \mathbb{R}^{Lg}$.
6. $\pi_\theta(a \mid o) = \dfrac{\exp\left(a\mathbf{z}(o)^T\theta\right)}{1+\exp\left(\mathbf{z}(o)^T\theta\right)}$ 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; Lo is the number of entries in the state vector O_i; Lf is the feature length of $f(O_i)$ after basic function; Lh is the feature length of $\mathbf{x}(f(O_i), A_i)$, which is the feature for the linear value approximate; Lg denotes the length of policy feature.

1 The algorithms for batch & online learning via LSTDQ & Fmincon

Suppose that $\mathbf{w}$ is the parameter for value function and θ is the policy parameter. In this section, we provide two optimization schemes:

- the batch learning for the given trajectory set. That is, for a given trajectory set, whose actions are randomly chosen, we learn an identical policy and value for all the N individual involved.
- the online learning for the new user in the future. The initial parameters $\widehat{\mathbf{w}}_0$ and $\hat{\theta}_0$ are given via the batch learning.

1.1 Batch learning for the Contextual bandit and discount reward

1. **Datasets via Micro-Randomized Trials**

   (a) Input parameters to generate the data (cf. Section 2.1)

   i. N the number of people involved in the data set.

   ii. T the trajectory length;

   iii. σ_r and σ_s are the noise strength when simulating the reward and the state respectively; σ_b is the variance of noise in the $\boldsymbol{\beta}$, which indicates how different the people are.

   (b) draw the trajectories for all the N people; each trajectory includes T time points. The simulation model is given in Section 2.1. Note that the action is randomly chosen.

   (c) 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

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

   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

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

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

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

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

2. **Algorithm for the batch RL via LSTDQ & Fmincon.**

   (a) **Input parameters** for the algorithm

i. ζ_c is the strength of ℓ_2-constraint on $\mathbf{w}$, which is the parameter for the value function.

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

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

(b) **Repeat**

i. ***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

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

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

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

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.

(c) **Until** convergence

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

3. **Appendix:** the gradient for (3) is as follows

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

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

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

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

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

1.2 Batch learning for the average reward method

1. **Datasets via Micro-Randomized Trials**

   (a) Input parameters to generate the data (cf. Section 2.1)

   i. N the number of people involved in the data set.

   ii. T the trajectory length;

   iii. σ_r and σ_s are the noise strength when simulating the reward and the state respectively; σ_b is the variance of noise in the $\boldsymbol{\beta}$, which indicates how different the people are.

   (b) draw the trajectories for all the N people; each trajectory includes T time points. The simulation model is given in Section 2.1. Note that the action is randomly chosen.

   (c) 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

$$\mathcal{D} = \{\mathcal{D}_n\}_{n=1}^N, \qquad \text{where } \mathcal{D}_n = \left\{\left(s_i, a_i, r_i, s_i^{'}\right) \mid i = 1, 2, \cdots, T\right\}. \tag{4}$$

   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

$$\begin{aligned}
\bar{\mathbf{X}} &= [\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2, \cdots, \bar{\mathbf{X}}_N] \in \mathbb{R}^{L \times NT} \\
\bar{\mathbf{Y}} &= [\bar{\mathbf{Y}}_1, \bar{\mathbf{Y}}_2, \cdots, \bar{\mathbf{Y}}_N] \in \mathbb{R}^{L \times NT} \\
\bar{\mathbf{r}} &= \left[\bar{\mathbf{r}}_1^T, \bar{\mathbf{r}}_2^T, \cdots, \bar{\mathbf{r}}_N^T\right]^T \in \mathbb{R}^{NT},
\end{aligned} \tag{5}$$

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

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

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

2. **Algorithm for the batch RL via LSTDQ & Fmincon.**

   (a) **Input parameters** for the algorithm

      i. ζ_c is the strength of ℓ_2-constraint on $\mathbf{w}$, which is the parameter for the value function.

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

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

   (b) **Repeat**

      i. ***Critic update to get*** *the optimal* $\widehat{\mathbf{w}}_0$ *as follows*

$$\widehat{\mathbf{w}}_0 = \left(\zeta_c \mathbf{I} + \frac{1}{NT} \sum_{i=1}^{NT} \left(\mathbf{x}_i - \bar{\mathbf{x}}_0\right)\left(\mathbf{x}_i - \gamma \mathbf{y}_{i+1}\right)^T\right)^{-1} \left(\sum_{i=1}^{t} \left(\mathbf{x}_i - \bar{\mathbf{x}}_0\right) r_i\right), \qquad (6)$$

      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; $\bar{\mathbf{x}}_0$ is he mean of value feature among all current states; $\mathbf{y}_{i+1} = \sum_a \mathbf{x}\left(f\left(S_{i+1}\right), a\right) \pi_\theta\left(a \mid S_{i+1}\right)$ 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.

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

$$0 = \sum_{i=1}^{t} \left(r_i - \eta + V\left(O_{i+1}; v\right) - Q\left(Q_i, A_i; \mathbf{w}\right)\right) \begin{pmatrix} 1 \\ \bar{O}_i\left(1 - A_i\right) \\ \bar{O}_i A_i \end{pmatrix} - \zeta_a \mathbf{I} \begin{pmatrix} 0 \\ \mathbf{w}_0 \\ \mathbf{w}_1 \end{pmatrix} \qquad (7)$$

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

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

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

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

Considring 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

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

■

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

$$\min_{\theta} \mathcal{J}(\theta) = -\frac{1}{NT} \sum_{i=1}^{NT} \sum_{a \in \{0,1\}} Q\left(S_i, a; \widehat{\mathbf{w}}_0\right) \cdot \pi_\theta\left(a|S_i\right) + \frac{\zeta_a}{2} \|\theta\|_2^2, \tag{11}$$

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.

(c) **Until** convergence

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

3. **Appendix:** the gradient for (11) 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

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

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

$$\begin{aligned}
\min_{\theta} \mathcal{J}_t(\theta) &= -\frac{1}{t} \sum_{i=1}^{t} \left(r_i + V\left(O_{i+1}; \widehat{\mathbf{w}}_t\right) - Q\left(O_i, A_i; \widehat{\mathbf{w}}_t\right)\right) + \frac{\zeta_a}{2} \|\theta\|_2^2, \qquad (12) \\
&= -\frac{1}{t} \sum_{t=1}^{t} \left(r_i + \sum_{a \in \mathcal{A}} Q\left(O_{i+1}, a; \widehat{\mathbf{w}}_t\right) \pi_\theta\left(a|O_{i+1}\right) - Q\left(O_i, A_i; \widehat{\mathbf{w}}_t\right)\right) + \frac{\zeta_a}{2} \|\theta\|_2^2 \\
&= -\frac{1}{t} \sum_{i=1}^{t} \left(r_i + \left(\mathbf{y}_{i+1} - \mathbf{x}_i\right)^T \widehat{\mathbf{w}}_t\right) + \frac{\zeta_a}{2} \|\theta\|_2^2 \qquad (13)
\end{aligned}$$

where $Q\left(O_i, A_i; \widehat{\mathbf{w}}_t\right) = \widehat{\mathbf{w}}_t^T \mathbf{x}_i$ and $V\left(O_{i+1}; \widehat{\mathbf{w}}_t\right) = \widehat{\mathbf{w}}_t^T \mathbf{y}_{i+1}$; $\mathbf{x}_i = \mathbf{x}\left(f\left(O_i\right), A_i\right)$ and $\mathbf{y}_{i+1} = \mathbf{x}\left\{f\left(O_{i+1}\right), \pi_\theta\left(1 \mid O_{i+1}\right)\right\}$. Considering the ℓ_2 penalty on θ and Matlab only has the minumizing algorithm, we get an equivalent minimizing problem as

$$\min_\theta \mathcal{J}_t\left(\theta\right) = -\frac{1}{t}\sum_{i=1}^{t}\left(R_i + \left(u_{i+1} - h_i\right)^T v\right) + \lambda\left\|\theta\right\|_2^2. \tag{14}$$

4. ***Gradient** for **actor update***: the objective function (12) is equivalent

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

Here we use an alternative updating method and treat $\widehat{\mathbf{w}}_t$ as a constant vector, rather than a vector function of θ; therefore, we have $\frac{d\widehat{\mathbf{w}}}{d\theta} = \mathbf{0}$. The gradient of (12) is

$$\nabla_\theta \hat{J}_t\left(\theta\right) = \frac{1}{t}\sum_{i=1}^{t}\sum_{a\in\{0,1\}} Q\left(O_{i+1}, a; \widehat{\mathbf{w}}_t\right)\cdot\left[\nabla_\theta \pi_\theta\left(a|O_{i+1}\right)\right] + \zeta_a\theta$$

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

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

where $a \in \{0, 1\}$ conreponds

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

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

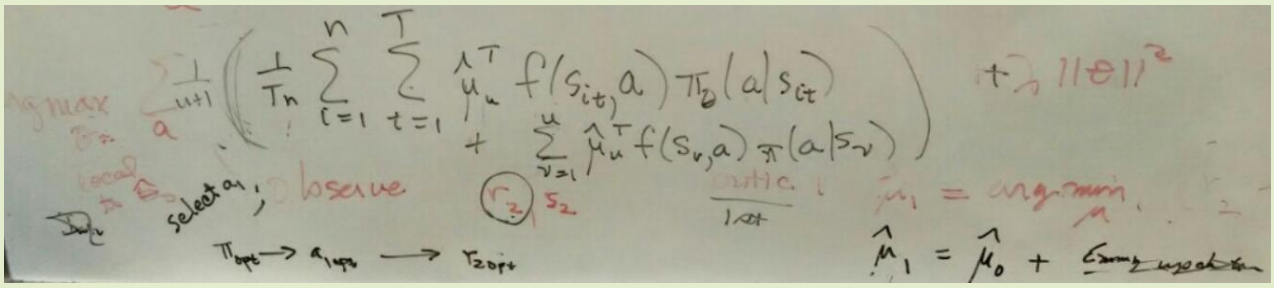


Figure 1: The objective function and solver we got in the meeting among Susan, Ambuj and Feiyun, on 06/14/2016.

1.3 Online RL learning for the actor-critic 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** ($t < T_{\max}$)

   (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*

$$\widehat{\mathbf{w}}_t = \mathbf{B}_t^{-1}\mathbf{a}_t, \tag{15}$$

   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 (15), resulting in the following equation

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

So at the beginning of the online learning process, i.e. $t = 1$, we have

$$\begin{aligned}
\widehat{\mathbf{w}}_1 &= \left(\mathbf{B}_0^{-1} - \frac{\mathbf{B}_0^{-1}\mathbf{x}\left(s_1, a_1\right)\mathbf{x}\left(s_1, a_1\right)^T \mathbf{B}_0^{-1}}{1 + \mathbf{x}\left(s_1, a_1\right)^T \mathbf{B}_0^{-1}\mathbf{x}\left(s_1, a_1\right)} \right) \left(\mathbf{a}_0 + \mathbf{x}\left(s_1, a_1\right) r_1 \right) \\
&= \mathbf{B}_0^{-1}\mathbf{a}_0 + \mathbf{B}_0^{-1}\mathbf{x}\left(s_1, a_1\right) r_1 - \frac{\mathbf{B}_0^{-1}\mathbf{x}\left(s_1, a_1\right)\mathbf{x}\left(s_1, a_1\right)^T \mathbf{B}_0^{-1}}{1 + \mathbf{x}\left(s_1, a_1\right)^T \mathbf{B}_0^{-1}\mathbf{x}\left(s_1, a_1\right)}\mathbf{a}_0 \\
&- \frac{\mathbf{B}_0^{-1}\mathbf{x}\left(s_1, a_1\right)\mathbf{x}\left(s_1, a_1\right)^T \mathbf{B}_0^{-1}}{1 + \mathbf{x}\left(s_1, a_1\right)^T \mathbf{B}_0^{-1}\mathbf{x}\left(s_1, a_1\right)}\mathbf{x}\left(s_1, a_1\right) r_1 \\
&= \widehat{\mathbf{w}}_0 + \mathbf{B}_0^{-1}\mathbf{x}\left(s_1, a_1\right) \left\{ r_1 - \frac{\mathbf{x}\left(s_1, a_1\right)^T \mathbf{w}_0 - r_1 + r_1 \left(1 + \mathbf{x}\left(s_1, a_1\right)^T \mathbf{B}_0^{-1}\mathbf{x}\left(s_1, a_1\right) \right)}{1 + \mathbf{x}\left(s_1, a_1\right)^T \mathbf{B}_0^{-1}\mathbf{x}\left(s_1, a_1\right)} \right\} \\
&= \widehat{\mathbf{w}}_0 + \mathbf{B}_0^{-1}\mathbf{x}\left(s_1, a_1\right) \left\{ \frac{\mathbf{x}\left(s_1, a_1\right)^T \mathbf{B}_0^{-1} - r_1}{1 + \mathbf{x}\left(s_1, a_1\right)^T \mathbf{B}_0^{-1}\mathbf{x}\left(s_1, a_1\right)} \right\}
\end{aligned}$$

where $\mathbf{B}_0$ and $\mathbf{a}_0$ come from the batch learning on the 1st set of people. In general, an incremental update at time point t is

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

(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 $\mathbf{w}_t$ in the value approximate.

1.4 Online RL learning for the 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.
2. **While** ($t < T_{\max}$)

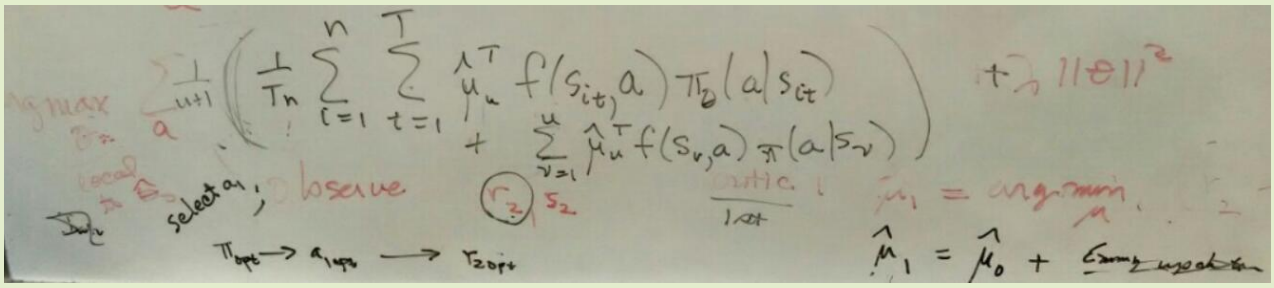


Figure 2: The objective function and solver we got in the meeting among Susan, Ambuj and Feiyun, on 06/14/2016.

(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 (cf. Section 1.1)*

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

where $\mathbf{x}_i = \mathbf{x}\left(f\left(O_i\right), A_i\right)$ is the feature vector at decision point i for the value funciton; $\mathbf{y}_{i+1} = \mathbf{x}\left\{f\left(O_{i+1}\right), \pi_\theta\left(1 \mid O_{i+1}\right)\right\}$ is the feature at next time point; r_i is the immediate reward at the i-th point; ζ_c is a 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

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

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 $\mathbf{w}_t$ in the value approximate.

1.5 Online RL learning for the average reward

1. **Input**:

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

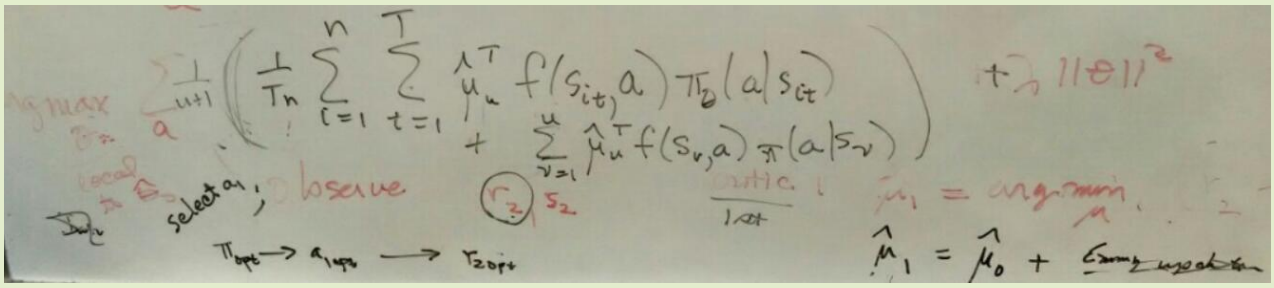


Figure 3: The objective function and solver we got in the meeting among Susan, Ambuj and Feiyun, on 06/14/2016.

(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.

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 *(cf. Section 1.2)*

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

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; $\bar{\mathbf{x}}_t$ is he mean of value feature among all current states; $\mathbf{y}_{i+1} = \sum_a \mathbf{x}\left(f\left(S_{i+1}\right), a\right) \pi_\theta\left(a \mid S_{i+1}\right)$ 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.

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

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

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.

2 The experiment settings

2.1 Datasets

2.1.1 #1 continuous simulation model

In the experiments, a trajectory for each individual is defined as follows

$$\mathcal{D}_T = \{(O_0, A_0, R_0), (O_1, A_1, R_1), \cdots, (O_T, A_T, R_T)\}, \tag{16}$$

where the initial states and action are generated by $O_0 \sim \text{Normal}_{p_1}\{0, \Sigma\}$ 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

$$\begin{aligned}
O_{t,1} &= \beta_1 O_{t-1,1} + \xi_{t,1}, && \text{weather-high is good} \\
O_{t,2} &= \beta_2 O_{t-1,2} + \beta_3 A_{t-1} + \xi_{t,2}, && \text{engagement} \\
O_{t,3} &= \beta_4 O_{t-1,3} + \beta_5 O_{t-1,3} A_{t-1} + \beta_6 A_{t-1} + \xi_{t,3}, && \text{treatment fatigue} \\
O_{t,j} &= \beta_7 O_{t-1,j} + \xi_{t,j}. \quad j = 4, \ldots, p
\end{aligned} \tag{17}$$

The corresponding immediate reward model is as follows

$$R_{t+1} = \beta_7 + A_t \times (\beta_9 + \beta_{10} O_{t,1} + \beta_{11} O_{t,2}) + \beta_{12} O_{t,1} - \beta_{13} O_{t,3} + \varrho_{t,}, \tag{18}$$

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 model (17) that is drawn from $\{\xi_{t,i}\}_{i=1}^p \sim \text{Normal}(0, \sigma_s^2)$. ϱ_t is the noise in the reward model (18), which is defined as $\varrho_t \sim \text{Normal}(0, \sigma_r^2)$.

2.1.2 #2 discrete simulation model (this part is from Peng Liao's proposal)

1. To verify the effect of the three acto-critic method, we consider a ***finite state*** and ***binary action MDP*** that models a dynamic system in mobile health setting. The action is coded to take values in $\{0, 1\}$, where $a = 0$ means "no treatment", and $a = 1$ means "providing an active treatment, e.g. sending an intervention to subject' s mobile device.
2. In the simulations, the subjects' trajectories $\{(S_0, A_0, R_1), (S_1, A_1, R_2, \cdots, (S_T, A_T, R_{T+1}))\}$ are i.i.d.

(a) The state variable $S_t = \{S_{t,1}, S_{t,2}, S_{t,3}\}$ is a ***three-dimensional vector*** with each element having finite values, i.e., $S_{t,i} \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$.

(b) The transition probability is given by

$$\Pr\left(S_{t+1,1} = s_1^{'}, S_{t+1,2} = s_2^{'}, S_{t+1,3} = s_3^{'} \mid S_{t+1,1} = s_1, S_{t+1,2} = s_2, S_{t+1,3} = s_3, A_t = a\right) = \prod_{i=1}^{3} p_i\left(s_i^{'} \mid s_i, a\right)$$

3. Each $p_i\left(s_i^{'} \mid s_i, a\right)$ is designed to model the dynamic system in the mobile health setting

   (a) the first element $S_{t,1}$ is interpreted as ***"weather variable"***;

   (b) the second element $S_{t,2}$ is ***"self-regulation"***;

   (c) the second element $S_{t,3}$ is ***"treatment fatigue"*** or ***"burden"***.

   In particular,

   (a) $S_{t,1}$ is ***not affected by actions***.

   (b) $S_{t,2}$ increases (decreases) instantly 立即 when (not) treated.

   (c) $S_{t,3}$ tends to stay around the same level when not treated

      i. and can only get higher when treated,

      ii. except for the highest level, e.g. we allow a small probability of returning to the second highest level.

4. The initial distribution of context is drawn from the uniform distribution, i.e $O_0 \sim$ uniform $(0,\ 1)$. The transition matrices are given as follows

$$p_1\left(j \mid i, a = 0\right) = p_1\left(j \mid i, a = 1\right) = \begin{bmatrix} 0.52 & 0.26 & 0.13 & 0.06 & 0.03 \\ 0.21 & 0.42 & 0.21 & 0.11 & 0.05 \\ 0.10 & 0.20 & 0.40 & 0.20 & 0.10 \\ 0.05 & 0.11 & 0.21 & 0.42 & 0.21 \\ 0.03 & 0.06 & 0.13 & 0.26 & 0.52 \end{bmatrix}$$

$$p_2\left(j \mid i, a = 0\right) = \begin{bmatrix} 0.80 & 0.20 & 0.00 & 0.00 & 0.00 \\ 0.62 & 0.31 & 0.07 & 0.00 & 0.00 \\ 0.38 & 0.38 & 0.19 & 0.05 & 0.00 \\ 0.28 & 0.28 & 0.28 & 0.14 & 0.03 \\ 0.22 & 0.22 & 0.22 & 0.22 & 0.11 \end{bmatrix}, p_2\left(j \mid i, a = 1\right) = \begin{bmatrix} 0.11 & 0.22 & 0.22 & 0.22 & 0.22 \\ 0.12 & 0.13 & 0.25 & 0.25 & 0.25 \\ 0.00 & 0.17 & 0.17 & 0.33 & 033 \\ 0.00 & 0.00 & 0.25 & 0.25 & 0.50 \\ 0.00 & 0.00 & 0.00 & 0.33 & 0.67 \end{bmatrix} \tag{19}$$

$$p_3\left(j \mid i, a=0\right)=\begin{bmatrix} 0.52 & 0.26 & 0.13 & 0.06 & 0.03 \\ 0.21 & 0.42 & 0.21 & 0.11 & 0.05 \\ 0.10 & 0.20 & 0.40 & 0.20 & 0.10 \\ 0.05 & 0.11 & 0.21 & 0.42 & 0.21 \\ 0.03 & 0.06 & 0.13 & 0.26 & 0.52 \end{bmatrix}, p_3\left(j \mid i, a=1\right)=\begin{bmatrix} 0.26 & 0.52 & 0.13 & 0.06 & 0.03 \\ 0.00 & 0.53 & 0.27 & 0.13 & 0.07 \\ 0.00 & 0.00 & 0.57 & 0.29 & 0.14 \\ 0.00 & 0.00 & 0.00 & 0.67 & 0.33 \\ 0.00 & 0.00 & 0.00 & 0.09 & 0.91 \end{bmatrix}.$$

5. The expected reward is given by

$$\mathbb{E}\left(R_{t+1} \mid S_t, A_t\right)=\beta_1 \times\left[\beta_2+\beta_3 S_{t,1}+\beta_4 S_{t,2}+A_t\left(\beta_5 S_{t,1}+\beta_6 S_{t,2}-\beta_7 \mathbb{I}_{S_{t,3}=1}\right)-\beta_8 g\left(S_{t,3}\right)\right] \tag{20}$$

where the most important parameter β is set as

$$\beta=[1000,1,0.4,0.2,0.2,0.2,0.05].$$

2.1.3 #3 the mixed simulation model of data #1 & #2

This data is a mixture of data #1 and data #2. We combine data #1's state transition model and data #3's immediate reward model.

2.1.4 #4 discrete simulation model by emphasizing delayed reward

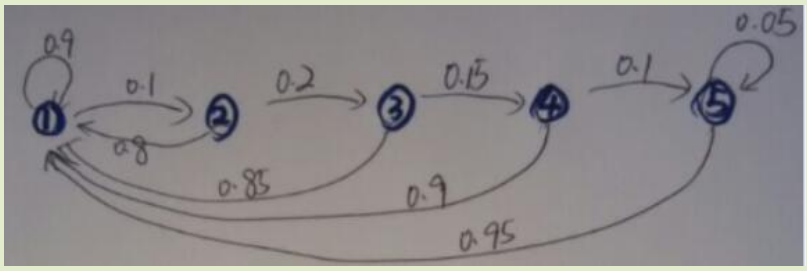
In this section, we design a simulation model that very emphasizes on the delayed reward. Compared with the discount reward method and average reward method, the contextual bandit method is supposed to perform badly on this data set. Specifically, the simulation data consists of the following aspects.

1. **State and actions**. Here we expect the simulation data to have 1-dimension state, $O_t \in \mathbb{R}$, with 5 value discrete value choices $O_t \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$. As usually, the action is set as binary, i.e. $A_t \in \{0, 1\}$.
2. **Transition probability**:

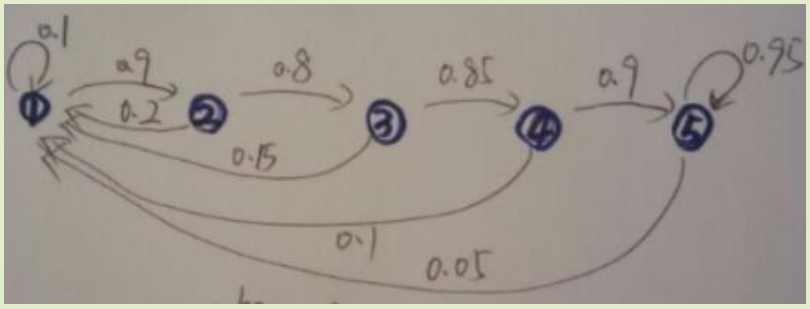
   (a) when $a=0$, the transition probability is set as follows

$$P\left(i \mid j, a=0\right)=\begin{bmatrix} 0.9 & 0.1 & 0 & 0 & 0 \\ 0.8 & 0 & 0.2 & 0 & 0 \\ 0.85 & 0 & 0 & 0.15 & 0 \\ 0.9 & 0 & 0 & 0 & 0.1 \\ 0.95 & 0 & 0 & 0 & 0.05 \end{bmatrix},$$

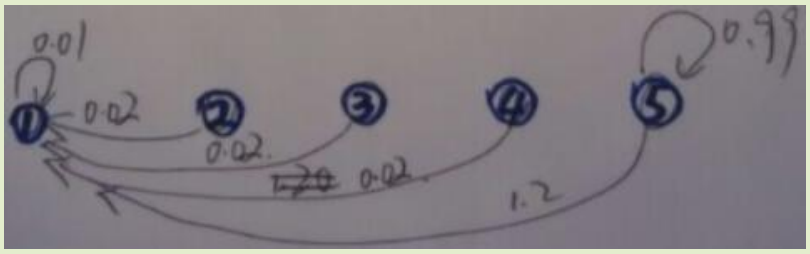
   which suggests the state to stays around $O_t=0.2$.



(a) The state transition probability when action $a = 0$.



(b) The state transition probability when action $a = 1$.



(c) The immediate reward model, which is a function of current state and next state.

Figure 4: The state transition probability and the immediate reward model for data #4, cf. Section 2.1.4.

(b) when $a = 1$, the transition probability is set as follows

$$P\left(i \mid j, a=1\right) = \begin{bmatrix} 0.1 & 0.9 & 0 & 0 & 0 \\ 0.2 & 0 & 0.8 & 0 & 0 \\ 0.15 & 0 & 0 & 0.85 & 0 \\ 0.1 & 0 & 0 & 0 & 0.9 \\ 0.05 & 0 & 0 & 0 & 0.95 \end{bmatrix}$$

which suggests the state to increase and stay around O_t =1.

3. **Immediate Reward**. Based on the talk with Ambuj on 02/02/2016, we design the reward model as follows

$$R\left(i \mid j\right) = \begin{bmatrix} 0.01 & 0 & 0 & 0 & 0 \\ 0.02 & 0 & 0 & 0 & 0 \\ 0.02 & 0 & 0 & 0 & 0 \\ 0.02 & 0 & 0 & 0 & 0 \\ 1.20 & 0 & 0 & 0 & 0.99 \end{bmatrix},$$

where the 1st dimension denotes the current state, and the 2nd dimension is the next state. Actually, the reward R_t is totally dependent on the states at the adjacent time points (i.e. independent on the action). Taking R_{t+1} for example, it is totally decided by the adjacent states O_t and O_{t+1}.

4. Since the bandit method only considers the immediate reward, it tends to choose the action $A_t = 0$, whose goal is to maximize the immediate reward. However for the methods with large γ, they emphasize the delayed reward, and get more long term average reward.

2.1.5 #5 continuous simulation model

This dataset is very simular with data#1. The only difference is that we introduce β_{14} to make it more like the heart steps application. After we have β_{14}, the immediate reward is usually between 1000 to 2000. In the experiments, a trajectory of tuples for each individual is defined as follows

$$\mathcal{D}_T = \left\{\left(O_0, A_0, R_0\right), \left(O_1, A_1, R_1\right), \cdots, \left(O_T, A_T, R_T\right)\right\}, \tag{21}$$

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

$$
\begin{aligned}
O_{t,1} &= \beta_1 O_{t-1,1} + \xi_{t,1}, && \text{weather-high is good} \\
O_{t,2} &= \beta_2 O_{t-1,2} + \beta_3 A_{t-1} + \xi_{t,2}, && \text{engagement} \\
O_{t,3} &= \beta_4 O_{t-1,3} + \beta_5 O_{t-1,3} A_{t-1} + \beta_6 A_{t-1} + \xi_{t,3}, && \text{treatment fatigue} \\
O_{t,j} &= \beta_7 O_{t-1,j} + \xi_{t,j}. \quad j = 4, \ldots, p
\end{aligned} \tag{22}
$$

The corresponding immediate reward model is as follows

$$
R_{t+1} = \beta_{14} \times \left[\beta_8 + A_t \times (\beta_9 + \beta_{10} O_{t,1} + \beta_{11} O_{t,2}) + \beta_{12} O_{t,1} - \beta_{13} O_{t,3} + \varrho_{t,}\right], \tag{23}
$$

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 (22) that is drawn from $\{\xi_{t,i}\}_{i=1}^{p} \sim \text{Normal}\,(0, \sigma_s^2)$. ϱ_t is the noise in the reward model (23), which is defined as $\varrho_t \sim \text{Normal}\,(0, \sigma_r^2)$.

To have a group of N people that is slightly different from each other, we need N slightly different MDPs. Formally, the $\boldsymbol{\beta}$ is used to differentiate the people by the following steps:

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

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

   where $\boldsymbol{\delta}_i \sim \text{Normal}\,(0, \sigma_b \mathbf{I}_{14})$ and $\mathbf{I}_{14} \in \mathbb{R}^{14 \times 14}$ is an identity matrix.
3. to make the future user, we will use this kind of method to generate a new $\boldsymbol{\beta}$.

2.2 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 RL results, i.e. the expectation of the long run average reward (i.e. ElrAR) $\mathbb{E}_\pi\left[\eta_{\hat{\theta}_t}\right]$ to verify the quality of the learnt policy. For each of the compared method (i.e. the contextual bandit, the discount reward and the average reward), a pair of parameters $\hat{\theta}_t$ (i.e. parameter in the policy model) and $\widehat{\mathbf{w}}_t$ (i.e. the parameter in the linear approximate model) will be learnt. To verify the quality of the learnt parameters $\hat{\theta}_t$ and $\widehat{\mathbf{w}}_t$, the ElrAR is used. Intuitively, ElrAR measures how many average reward in the long run we could get by using the learnt policy $\hat{\theta}_t$. A larger ElrAR corresponds to a better performance. Formally, there are three steps to get the ElrAR:

- For each individual, a trajectory of length $T = 10,000$ will be generated by the using the policy parameter $\hat{\theta}_t$;

- By averaging the rewards for the last $9,000$ elements from the trajectory, we could get the long run average reward

$$\eta_{\hat{\theta}_t} = \frac{1}{T-i}\sum_{j=i}^{T} R_{j+1}$$

where $i = 1000, T = 10000$.

- The expectation of the long run average reward is obtained by averaging $\eta_{\hat{\theta}}$ among all the N individual results the as

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

2.3 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, 3].

2.3.1 The polynomial basis [1]

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

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

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:

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

2.3.2 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 characteristic】
   (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.

[3] ——, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2012.