Learning Dynamic Treatment Strategies by Deep RL

Elynn Y. Chen

March 21, 2019

1 Overview

For the final project, I plan to use deep reinforcement learning algorithms to develop a decision support tool that leverages available patient information in the data-rich intensive care unit (ICU) setting. The specific dynamic decision problem I am considering is the management of invasive mechanical ventilation, and the regulation of sedation and analgesia during ventilation. The following sections will discuss in detail the background of the problem, the formulation of dynamic treatment decision as in a RL framework, and candidate RL algorithms to solve this problem. The **resources** to be used contains two part, namely, the medical data and computing resource. For the medical data, I will use the MIMIC-III - a freely accessible critical care database (Johnson et al., 2016). For computation, I will use AWS. My **mentor Lilian** has rich experiences in an array of deep reinforcement learning algorithms. She would advice me on which deep RL algorithm to use and how to adapt algorithms to fit specific application problems. A **tentative schedule of deliverables** are shown in Table 1.

Week	Deliverables
Mar. 25 - 29	Dataset and simple IO scripts.
Apr. 1 - 5	Q-Learning, Fitted Q-Learning and DQN, which are used mostly by current literatures.
Apr. 8 - 12	Double Q / Dueling
Apr. 15 - 19	Advanced off-policy algorithms such as SAC
Apr. 22 - 26	Writing report and prepare slides

Table 1: My tentative schedule of deliverables.

2 Medical Online Decision Problem

The problem of treatment in medical condition is a online decision problem. For example, as illustrated in Figure 1, faced with a patient with sepsis, the intensivist must decide whether and when to initiate and adjust treatments such as antibiotics, intravenous fluids, vasopressor agents and mechanical ventilation.

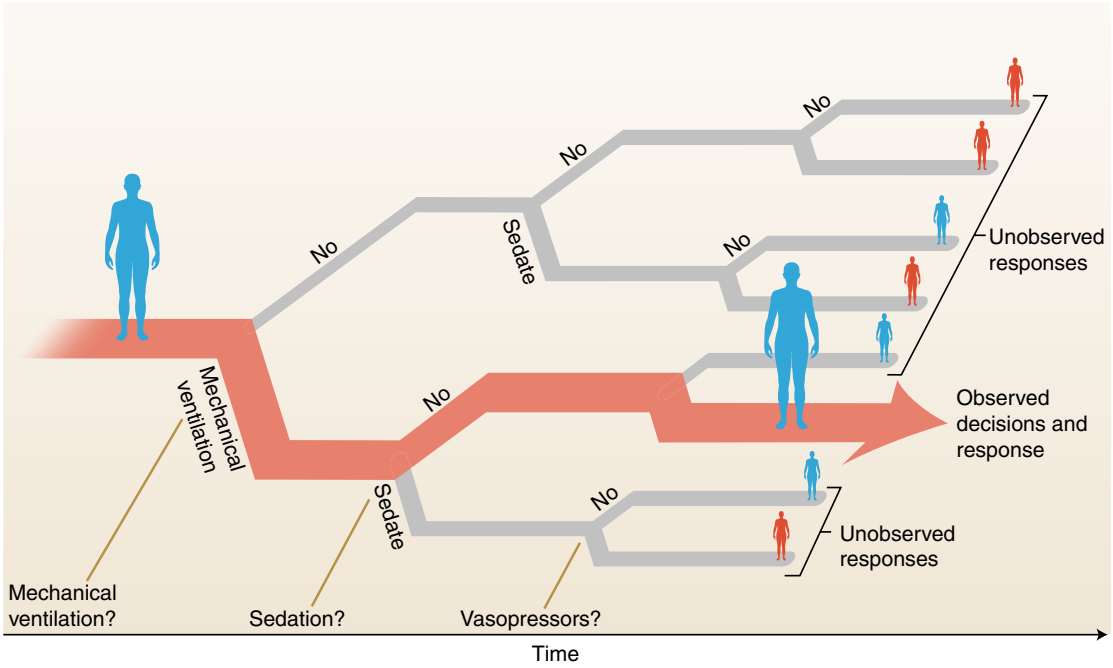

Figure 1: Sequential decision-making tasks. Image obtained from Gottesman et al. (2019). Credit: Debbie Maizels / Springer Nature

Currently, these decisions are based on heuristic. The widespread adoption of electronic health records (EHRs) opens new opportunity for a data-driven approach to health care. Potentially, aid doctors in decision making.

Reinforcement learning (RL) provides tools to optimize sequences of decisions for long-term outcomes. The aim of this project is to develop a decision support tool that leverages available patient information in the data-rich ICU setting to alert clinicians when a patient is ready for initiation of weaning, and to recommend a personalized treatment protocol.

3 Critical Care Data

The data set we use is the Multi-parameter Intelligent Monitoring in Intensive Care (MIMIC III) database (Johnson et al., 2016) – a freely available source of de-identified critical care data

from 53,423 adult admissions and 7,870 neonates. The data includes patient demographics, time-stamped measurements from bedside monitoring of vitals, administration of fluids and medications, results of laboratory tests observations and notes charted by care providers, as well as diagnoses, procedures and prescriptions for billing.

4 The RL Approach to the Dynamic Treatment Decision

The dynamic treatment decision problem can be formulated as the problem of learning solution of an optimal control problem from a sample of trajectories. We consider the discrete-time optimal control problems for which the aim is to maximize a sum of discounted rewards over an infinite time horizon.

4.1 MDP Formulation

A Markov Decision Process is characterized by:

- A finite **state space** $\mathcal{S}$: at each time t, the patient is in state $s_t \in \mathcal{S}$.
- An **action space** $\mathcal{A}$: at each time t, the agent takes action $a_t \in \mathcal{A}$ that changes the current state to next state s_{t+1}.
- A **transition function** $P(s_{t+1}|s_t, a_t)$ is the probability of the next state given the current state and action taken. It defines the (unknown) dynamics of the system.
- A **reward function** $r(s_t, a_t) \in \mathbb{R}$, the observed feedback following a transition at each time step t.

We define the corresponding elements in out ICU mechanical ventilation weaning application. The state s_t is a 32-dimensional feature vector including fixed demographic information (age, weight, gender, admit type, ethnicity), relevant physiological measurements, ventilator settings, level of consciousness (measured by Richmond Agitation Sedation Scale, or RASS), current dosages of different sedatives, time into ventilation, and the number of intubations so far in the admission.

The action a_t is a 2-dimensional vector, of which the first entry $a_t[0] \in \{0, 1\}$ specifies whether the patient be off or on the ventilator, respectively, and the second $a_t[1] \in \{0, 1, 2, 3\}$ indicates the level of sedation to be administrated over the next 10-minute interval. Thus, there are 8 possible actions in total. Note that, in medical practice, there are six commonly used sedatives and the

dosage scales varies. Here, for simplicity, we map the six commonly used sedatives approximately into a single dosage scale, and discretize it to four different levels of sedation.

Note that the above simplification of state and action space is only a first attempt. We expect to expand them to allow more complicated setting later.

The reward signal is important and need crafted carefully in real applications. In this application, we build our reward signal based on (i) physiological stability, (ii) failed SBTs or reintubation, and (iii) time into ventilation.

4.2 Learning the Optimal Policy

In this application, we plan to focus on the off-policy algorithms. Specifically, we aim at estimating the expected value of state-action pairs $Q_\pi(s,a) : S \times A \to \mathbb{R}$. The best policy will be determined based on $\widehat{Q}_\pi(s,a)$. An off-policy algorithm starts with an initial state and arbitrary approximation of the Q-function, and update this estimate using the reward from the next transition using the Bellman update:

$$\widehat{Q}(s_t,a_t) = \widehat{Q}(s_t,a_t) + \alpha(r_{t+1} + \gamma \max_{a\in\mathcal{A}}(\widehat{Q}(s_{t+1},a) - \widehat{Q}(s_t,a_t))),$$

where the learning rate α gives the relative weight of the current and previous estimate, and γ is the discount factor.

The most common methods in current literature are Q-learning and fitted Q-iteration. I will first try those methods on the data set. Further, I will explore more sophisticated methods. At last I will compare results from different methods.

References

Gottesman, O., F. Johansson, M. Komorowski, A. Faisal, D. Sontag, F. Doshi-Velez, and L. A. Celi (2019). Guidelines for reinforcement learning in healthcare. *Nature medicine 25*(1), 16–18.

Johnson, A. E., T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark (2016). Mimic-iii, a freely accessible critical care database. *Scientific data 3*, 160035.