Learning a policy for the management of children managed with ECMO: a reinforcement learning approach

Omeed Ashtiani, [1] **Harsha Kokel,** [1] **Michael A. Skinner,**[1]
[1]University of Texas at Dallas, Dallas TX

Abstract

ExtraCorporeal Membrane Oxygenation (ECMO) is a method of supporting patients with life-threatening respiratory or cardiac failure. In an effort to increase the efficacy of the ECMO treatment, we utilize a Reinforcement Learning (RL) technique known as Batch Reinforcement Learning (Batch RL) which proves especially useful in situations limited to offline data. Extracting physician action data from Electronic Health Records (EHR), our goal is to discover an optimal policy via Fitted Q-Learning, a type of Batch RL, from the offline patient trajectories in order to optimize the set of actions available at each state of the ECMO process. While there is currently a lack of variability in the Q-values for different actions in their respective states, we believe this work opens the door to explore future avenues by methods such as synthesizing trajectories, reward shaping, and action filtering.

The explosion of medical data available in the electronic health record (EHR) allows increasingly fruitful automated patient-based research using techniques of artificial intelligence (AI) and machine learning (ML). These modalities have had particularly remarkable success in evaluating and classifying radiographic and other varieties of images (Topol 2019). Other successes include the elicitation of rule-based treatment strategies from medical records (Skinner et al.), ascertaining the need for cardiac procedures (Yang et al. 2017), predicting drug success in the management of mental illness (Chekroud et al. 2016), among others (Yu, Liu, and Nemati 2019).

Another promising area through which ML and AI techniques contribute to improved medical care is in the automatic derivation of a patient management policy from the EHR, which could be used to improve health care in a number of ways. For example, we could automatically discover optimal policies for managing particular diseases. Moreover, an optimal policy, once discovered, could be compared to a patient's actual clinical course; if there is a deviation, physicians could be provided with suggestions for care. Finally, the ability to extract medical polices from EHRs would enable predictions of patient prognosis and outcomes.

Reinforcement learning (RL) is an AI technique for sequential decision making in which an agent explores a space of states, taking actions and receiving rewards, aiming to find the optimal policy mapping actions to states. This technique has been used in many medical applications (Yu, Liu, and Nemati 2019). Moreover, in some cases, it appears that the learning agent may be able to learn a policy that is superior to most of the policies that were followed by the physicians. (Komorowski et al. 2018).

A challenge in the extraction of physician actions from the medical record using RL is that an agent cannot explore the state space to investigate which actions are optimal. Rather, we must discover the best policy from a set of patient trajectories, with the hope that the EHR information is adequate to optimize over the set of actions available at each state. Thus, we must implement the RL technique known as "Batch reinforcement learning", which will be described below (Ernst, Geurts, and Wehenkel 2005; Lange, Gabel, and Riedmiller 2012).

Methods

Exracorporeal membrane oxygenation

Extracorporeal membrane oxygenation (ECMO) is a method of supporting patients with life-threatening respiratory or cardiac failure. The technique requires surgical placement of large cannulas in the neck or in the heart, and externally circulating the patient's blood through a system that oxygenates the blood and removes carbon dioxide. Reserved for the most critically ill of patients, ECMO mortality can be very high and even among survivors there are frequent treatment complications (Lin 2017).

Patient data

We used de-identified medical data abstracted from EHRs for 140 children treated at the Children's Medical Center of Dallas who survived their period of ECMO. The study was performed in accordance with an exemption granted by the University of Texas Southwestern Institutional Review Board (IRB). The time on ECMO ranged from 6 to 985 hours, averaging 174 hours. For each hour of ECMO bypass, and for from 1 to 24 hours prior to cannulation (15 hours, on average), 40 physiologic and laboratory parameters were recorded. Not every parameter was measured each

Table 1: Study parameters.

Parameter	Units
Mean arterial pressure	mm Hg.
Heart rate	beats/min
Respiratory rate	breaths/min
pH	none
pO2	mm Hg.
Pressure volume sensor	cm H2O
Measured flow	ml/kg-min

hour; for example, those exclusively associated with ongoing bypass (such as pump flow) were only recorded while the child was actually undergoing ECMO support.

The children underwent brain imaging after their period of bypass to monitor for intracranial bleeding or stroke. In 74 cases (53 percent), a moderate or severe intracranial injury was identified. Our overarching aim is to discover an ECMO management policy that reduces the risk of such injuries.

We chose seven physiologic parameters thought to be the most useful for managing the respiratory and hemodynamic status of patients. These are tabulated, with the units of measurement, in Table 1. For our analysis, we abstracted the actual physiologic values to take one of three values at each time point denoting whether the value was in the normal range, significantly decreased, or significantly elevated.

For use in the Batch RL algorithm, we extracted from the patient trajectories 32215 $\langle SARS' \rangle$ tuples, representing an initial state, action, reward and subsequent state. For each tuple, reward shaping was performed by rewarding actions that tended to normalize the patient's physiologic state, and penalizing those that resulted in a more abnormal state. A final positive reward was assigned at the end of the trajectory if the child exhibited normal brain scans, and a negative reward otherwise.

Fitted Q-iteration for batch learning

Batch RL is a sub-field of RL, traditionally used to learn an optimal policy in the setting where we lack a simulator or an explorable environment. In this environment, the learning agent must extract a policy from a fixed number of trajectories given *a priori*. Before explaining fitted-Q iteration for Batch RL, we motivate the need for this algorithm by providing a brief background on Q-learning and highlight its drawback for batch RL.

The Bellman optimality equation for the action-value function (Q) is given as:

$$Q^*(s,a) = \sum_{s'} T(s,a,s') \Big[R(s,a,s') + \gamma \max_{a'} Q^*(s',a') \Big], \tag{1}$$

where $T(s,a,s')$ is a transition probability of landing in state s' on taking action a in state s and $R(s,a,s')$ is a Reward at state s' reached on taking action a in state s'

In the dynamic programming, the above equation is implemented as:

$$Q_{k+1}(s,a) \leftarrow \sum_{s'} T(s,a,s') \Big[R(s,a,s') + \gamma \max_{a'} Q_k(s',a') \Big] \tag{2}$$

Q-Learning is a model-free approach to learn the Q values by exploring the environment, i.e. performing actions based on some policy. A table of Q values for each state action pair, $Q(s,a)$, is maintained and the table is updated at every step using the running average formula:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + (\alpha) \Big[R(s,a,s') + \gamma \max_{a'} Q(s',a') \Big] \tag{3}$$

One major drawback of Q-learning is **exploration-overhead**, as Q value update is made after every action, it needs lot of trajectories for convergence. Since in Batch RL we have a fixed number of trajectories, we need to find a more efficient algorithm.

When using function approximations, Q values are estimated by a function and after every action the function is updated using following equation:

$$f(s,a) \leftarrow (1-\alpha) f(s,a) + \alpha \left(r + \gamma \max_{a' \in A} f(s',a') \right) \tag{4}$$

This asynchronous update in function approximation attempting to improve the Q value of a single state after an action might impair all other approximations. This **inefficient approximation** is another major disadvantage of Q-learning algorithms.

Gordon 1995 provided a stable function approximation approach by separating dynamic programming step from function approximation step. Effectively, He split the above function update equation to two steps.

$$\begin{aligned} f'(s,a) &\leftarrow r + \gamma \max_{a' \in A} f(s',a') \quad \forall s,a \\ f(s,a) &\leftarrow (1-\alpha) f(s,a) + \alpha f'(s,a) \end{aligned} \tag{5}$$

He proved that this approximation is guaranteed to converge and will result in better approximation.

Ernst, Geurts, and Wehenkel 2005 proposed fitted Q iteration by borrowing the splitted approach from Gordon. The approach proposes iterative approximation of Q value by reformulating the Q-Learning as a supervised regression problem. Algorithm 1 shows procedure for fitted-Q learning.

(Ernst, Geurts, and Wehenkel 2005) proposed regression tree as a function approximators, but since then various different approximation functions have been used. For this project, we also focus on learning regression trees. We ran the Fitted-Q learning algorithm for 100 iterations, after which the trees and Q-values had converged.

Algorithm 1 Fitted-Q Iteration

INPUT: tuples $< s, a, r, s' >$, γ, `stopping condition`
OUTPUT: $Q(s, a)$

function `Fitted` $-$ `Q`
  $Q(s, a) = 0$
  while `not stopping condition` **do**
    ▷ Generate regression dataset
    $\mathbf{X} =< s, a >$ ▷ features
    $Y = r + \gamma max_{a'} Q(s', a')$ ▷ regression values
    $Q(s, a) = learn(\mathbf{X}, Y)$
    ▷ Fit a regression function on the dataset
  end while
  return $Q(s, a)$
end function

Results

It is instructive to examine a list of Q-values for a representative state. Notably, there is not much variability in the expected value that accrues following the different actions, as seen in . This suggests that acting greedily at each state (to improve the physiologic status of the patient) is not associated with a significant gain in expected value.

895.54	808.45	895.54	895.54	810.45	895.54
895.54	895.54	895.54	895.54	804.18	998.55
895.54	808.65	895.54	895.54	895.54	

Table 2: Vector of Q values for different actions and state: $\langle -1, 0, 0, 0, 0, 0, 0 \rangle$

To formally evaluate how well we were able to encourage the agent to select the best action greedily, we computed the mean reciprocal rank (MRR) of the optimal action for each state, obtained from a domain expert. The MRR values obtained for each of the actions is presented in Table . Interestingly, despite the fact that the expected rewards do not appear to vary significantly across the possible actions, for most of the actions, the optimal action ranked between 2 and 3 among the 17 action choices.

Two of the possible actions, "cannulate" and "decannulate" occur rarely- once each per trajectory. To investigate how the exclusion of these uncommon actions might affect our results, we rebuilt the model and recomputed the MRR values, shown in figure 1 .Interestingly, the MRR values demonstrate marked improvement.

Discussion and future directions

This preliminary work suggests a number of future avenues of research. In our experiments, we used domain-expertise to assign a reward value to each action in the patient trajectory, rewarding actions that tend to move the patient's state in the direction of normal physiology. It is interesting to note

Action	Mean Reciprocal Rank
Heart Rate decr	0.5462
Heart Rate incr	0.5003
MAP decr	0.599
MAP incr	0.4313
Measured Flow decr	0.6646
Measured Flow incr	0.4071
pH decr	0.5238
pH incr	0.6486
pO2 decr	0.9048
pO2 incr	0.7734
Pressure Volume Sensor decr	0.4266
Pressure Volume Sensor incr	0.5355
Resp Rate decr	0.5971
Resp Rate incr	0.6944
No Action	0.2
Overall	**0.5916**

Table 3: Mean Reciprocal Rank (MRR) for each action and over all the actions.

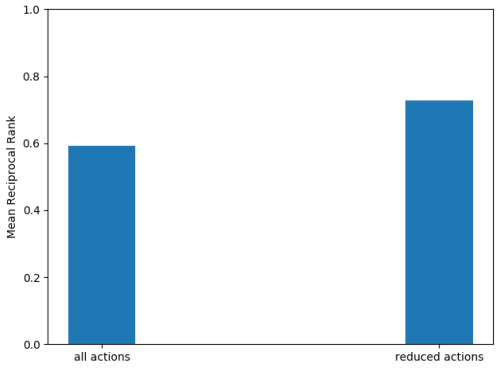

Figure 1: Mean Reciprocal rank comparison: including all actions vs reduced actions by removing the "cannulate" and "decannulate".

that when we look at the Q-values in the model, there is very little discrimination in expected reward among the 17 possible actions; the expected reward is pretty constant among all of the states, over all of the actions. It is possible that the data are simply non-informative, and are not amenable to our policy discovery aim. Alternatively, alterations in our hand-crafted reward shaping may improve our predictions- perhaps by assigning higher rewards to actions tending to correct more important physiologic aberrations. We will also see how the results change in the absence of any rewarding during the trajectories, relying only on the reward based on the final state of the patient.

We noted above that the removal of the "cannulate" and

"decannulate" actions seems to improve the MRR values, suggesting an increased correlation between the best greedy action and the expected reward over the entire trajectory. To further investigate this phenomenon, we will consider excluding the states prior to placing the patient on ECMO bypass, thereby modeling the patient course only after cannulation has occurred.

Finally, having extracted the Q-values for each state and action, we are in a position to synthesize trajectories by sampling our initial states, and then taking the action at each state maximizing the Q-value. We will thereby determine whether such a regime increases the expected value of the start state.

References

Chekroud, A. M.; Zotti, R. J.; Shehzad, Z.; Gueorguieva, R.; Johnson, M. K.; Trivedi, M. H.; Cannon, T. D.; Krystal, J. H.; and Corlett, P. R. 2016. Cross-trial prediction of treatment outcome in depression: a machine learning approach. *The Lancet Psychiatry*.

Ernst, D.; Geurts, P.; and Wehenkel, L. 2005. Tree-Based Batch Mode Reinforcement Learning. *Journal of Machine Learning Research*.

Gordon, G. J. 1995. Stable Function Approximation in Dynamic Programming. *ICML*.

Komorowski, M.; Celi, L. A.; Badawi, O.; Gordon, A. C.; and Faisal, A. A. 2018. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nature Medicine*.

Lange, S.; Gabel, T.; and Riedmiller, M. 2012. Batch reinforcement learning. In *Adaptation, Learning, and Optimization*.

Lin, J. C. 2017. Extracorporeal membrane oxygenation for severe pediatric respiratory failure. *Respiratory care*.

Skinner, M. A.; Raman, L.; Shah, N.; Farhat, A.; and Natarajan, S. Elicitation of probabilistic logic rules from medical records: A preliminary study in learning policies for the management of critically ill children (presented at 2019 conference on probabilistic logic programming, las cruces nm).

Topol, E. J. 2019. High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine*.

Yang, S.; Hadiji, F.; Kersting, K.; Grannis, S.; and Natarajan, S. 2017. Modeling heart procedures from ehrs: An application of exponential families. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*.

Yu, C.; Liu, J.; and Nemati, S. 2019. Reinforcement learning in healthcare: A survey. *arXiv preprint arXiv:1908.08796*.