AA228 Final Project: Saving Artificial Intelligence Clinician

Andriy Syrov
asyrov@stanford.edu

Robert P. Trevino
rptrevin@stanford.edu

Sergio Gonzales
checo@stanford.edu

***Abstract*—The use of Reinforcement Learning (RL) to aid clinician in screening, resource allocation, and treatment decision has grown substantially in the last few years. Komorowski et al. [1] is one notable work that developed a POMDP to model decision making when treating Sepsis. This work promises higher performance of AI algorithm compared to clinician, yet limitations of this algorithm are then uncovered and described in work by Jetter et al. [2]. In our work, we reproduce the works of Komorowksi et al. and explore methods that address its limitations as described by Jetter et al. We implement alternatives to state representation and alternative methods of learning: Q-learning and SARSA with Value Function Approximation (VFA). We find that Q-learning produced the same results as Policy Iteration, which was implemented by Komorowski et al., but SARSA with VFA learned more nuanced policies based on survival status.**

I. Introduction

Sepsis, a systemic infection that results in significant end organ dysfunction (e.g. kidney failure, brain damage) due to inadequate blood flow, is one of the most common life-threatening conditions treated in an intensive care unit (ICU) in the United States. It is notoriously difficult to manage and as a consequence there is great intra-clinician variability in treatment practices: research that can help improve guidelines for clinicians are of great interest to the medical community. Many researchers have attempted to develop decision making models for clinicians [3]–[6] most notably Komoroski et al., which was published in *Nature* in 2018 [1]. For our project we attempt to reproduce the findings of Komoroski et al., and then implement several techniques to address its limitations.

A. Previous Research

In their 2018 paper, Komoroski et al. [1] introduce a RL algorithm method to learn treatment policies for Sepsis patients. In their setting, states are characterized by 48 discrete and continuous variables such as demographics and physiological measurements. The state space is reduced to 750 discrete clusters of patient characteristics found with kmeans++ algorithms. States are defined for 4-hr non-overlapping time intervals and contains up to 72 hours of measurements.

To treat Sepsis, the patient's circulatory system is supported through administration of a vasopressors (a drug) or intravenous (IV) fluids. The set of actions is defined as the cross product of a set of discretized doses of a vasopessor and a set of discretized IV fluid volumes. Finally, the authors use Policy Iteration to solve the model.

While the authors show that the average policy value (of 500 policies) is greater than the average value of the actions taken by clinicians, there is substantial variation policy performance. In fact, distribution of the models' performance is not statistically different than the no-drug policy or a random set of actions as a policy. This and other limitations of this study are well described by a 2019 pre-print by Jetter et al. [2]. We aim to address issues with state representations that are biased toward healthy patients and methods of solving policies that fail to achieve clinician performance.

B. Novel approaches

In addition to replicating Komoroski et al. [1], we bring several novel contributions to existing models that suggest and optimal treatment policy for Sepsis patients using Reinforcement Learning. We am to address shortcomings of existing models in areas of:

1) Patient state representation
2) Learning procedures without health care domain insights
3) Failure to find policies consistent with correct clinical treatments

We implement an alternative approach to clustering patient states with a Variational Auto-Encoder (VAE), we also employ Q-learning as method for solving for a policy, and we expand state representation to a continuous space by using SARSA with Value Function Approximation (VFA).

As in Komorowski et al. [1], we used the restricted-use Medical Information Mart for Intensive Care III (MIMIC-III) [7] data set, which contains over 60,000 intensive care unit admission records in 2 large Boston hospitals and over 300 million chart events that include rich features such as demographics, laboratory tests, diagnoses, procedures, etc. for our decision making modeling task under state, outcome, measurement, and interaction uncertainty. All members of the group complete the required training and use agreements to gain access the data. The data was made accessible via BiqQuery in Google Cloud, where analysis, queries, and tables were executed and defined.

II. Formulation

A. Markov Decision Process Formulation

State: We approach the problem in discrete state space setting and continuous. In **discrete setting** we have set of states $S = \{s_1, s_2, ...s_n\}$ with $n = 750$ number of states.

Each state obtained from clustering of continuous $\mathbb{R}^{48}$ patient features space (defined as demographics, physiological conditions such as heart rate, respiratory rate, Mean Arterial Pressure [MAP], arterial pH, etc.). In **continuous setting** we join several states into single $\mathbb{R}^{n \cdot k}$ vector for neural network value function approximation. Joining $k = 2$ last states allows algorithms capture some more of patient dynamics, such as decreasing heart rate or change in arterial pressure. The algorithm executes an action at fixed 1-hour intervals and transitions to new state. Last state is terminal and corresponds to patient discharge.

Action: Our actions is set of treatments types to patient. Let $A = \{a_1, a_2, ..a_m\}$ be the actions that can be taken on a patient in a given state s, defined as the discretized doses of intravenous fluids and vasopressor into 5 bins each. This produces a combination of 25 different actions that can be taken. The reward function, $R : S \times A \rightarrow \mathbb{R}$ is defined as survival of sepsis 90 days post ICU admission for patients suffering from sepsis.

Reward: We use sparse reward, with single value of -1 for deceased patients and 1 for survived patients. This reward is received when transitioning to terminal state. The utility function is then defined as:

$$U(s_1) = \sum_{t=1}^{n} \gamma^{t-1} r_t$$

where s_1 is initial patient state at admission time, n is number of decisions before patient discharges, r_t is discounted by $\gamma = 0.999$ rewards, and r_t is immediate reward at time step t.

Objective: The objective function then becomes choosing optimal policy that maximizes the expected reward function from patient start state to final discharge from hospital (horizon):

$$\pi^*(s) = \arg\max_{\pi} U^{\pi}(s)$$

III. Algorithms

In this section, we provide an overview of the various algorithms we applied to the problem. The experiment setup and results are described in Section IV. The algorithm information for both Policy Iterator and Q-Learning are directly from the course text book by Kochenderfer et al. [8].

A. Space Clustering

For space clustering we used 2 approaches. First approach is to directly cluster $\mathbb{R}^{48}$ state vector space using kmean++ algorithm. For the second approach we used a VAE to reduce space dimensionality to $\mathbb{R}^3$ and then clustered that space. Both algorithms showed similar performance, which described in more details in section IV.

B. SARSA with Value Function Approximation

SARSA is variation of Q-learning algorithm. It is on-policy algorithm, which is in our case follows clinician policy while trying to learn a better, more optimal policy assuming clinician does reasonable exploration. We are running this algorithm in an offline setting. Value Function Approximation (VFA) was implemented as neural network with input vector of size $\mathbb{R}^{96}$ (current and previous feature vector of patient conditions) and output vector $\mathbb{R}^{25}$ (value for each medication treatment action). Hidden layers were implemented with ReLU activation function.

Algorithm 1 SARSA with VFA

1: **for** $iteration = 1, 2, 3, ..., T$ **do**
2: Observe batch $B = [(s, a, r, s_{next}, a_{next}), ...]$
3: **for all** $(s, a, r, s_{next}, a_{next}) \in B$ **do**
4: target $\leftarrow$ predict($Q_\theta[s]$)
5: target[a] $\leftarrow r + \gamma \cdot$predict($Q_\theta[s_{next}]$)[$a_{next}$]
6: fit($Q_\theta[s]$, target)
7: **end for**
8: **end for**

(Note: target, line 4, is vector in $\mathbb{R}^{25}$, line 5 updates a element of this vector. Function predict(.) also returns $\mathbb{R}^{25}$ vector). We trained algorithm on 4866 or 5366 patients and used the rest 500 patient trajectories for evaluation.

C. Policy Iteration

Policy iteration was investigated due to its simplicity and use by Komoroswki et al. [1]. As shown in Algorithm 2, it is implemented using iterative evaluation step shown in Algorithm III-C and policy improvement through the look-ahead function shown in Algorithm 4. The gamma value was set to 0.95 and the total number of iterations was 200.

Algorithm 2 Policy iteration

$\mathcal{S}, \pi = \mathcal{P}.\mathcal{S}, M.\pi$ {$\mathcal{P}$ is an MDP data structure and M contains policy information}
U=[0.0 for s in $\mathcal{S}$]
for k in 1:k_{max} **do**
U= PolicyIterativeEvaluation(P)
π' = ValueFunctionPolicy($\mathcal{P}, U$)
if all $\pi(s) == \pi'(s)$ for s in $\mathcal{S}$ **then**
break
end if
$\pi = \pi'$
end for
return π

The iterative policy evaluation step allows for the policy to be evaluated using the look-ahead function.

Algorithm 3 PolicyIterativeEvaluation

$\mathcal{S}, T, R, \gamma = \mathcal{P}.\mathcal{S}, \mathcal{P}.T, \mathcal{P}.R, \mathcal{P}.\gamma$ {$\mathcal{P}$ is an MDP data structure}
U=[0.0 for s in $\mathcal{S}$]
for k in k_{max} **do**
U=[lookahead($\mathcal{P}$,U,s $\pi(s)$) for s in $\mathcal{S}$]
end for

The look-ahead function plays a pivotal roll in both evaluation as well as improvement steps of the policy iterator algorithm.

Algorithm 4 Lookahead

1: $\mathcal{S}, T, R, \gamma = \mathcal{P}.\mathcal{S}, \mathcal{P}.T, \mathcal{P}.R, \mathcal{P}.\gamma$ $\{\mathcal{P}$ is an MDP data structure$\}$
2: return R(s,a) + γ sum(T(s,a,s')*U(s') for s' in $\mathcal{S}$)

At the end of the policy iteration, there were still states that did not have actions associated with them. Assuming a power law distribution of actions, a count was performed to determine which action was predominant across the different states. This action was then used as the default action for those states that did not have any information in the file.

D. Q-Learning

We also investigated the Q-Learning algorithm, which relies on a state-action pair for analysis. The algorithm updates utility values associated with state-action pairs to find an approximate solution. The update of the utility value associated with the state-action pairs is defined as

$$Q(s,a) = Q(s,a) + \alpha(r + \gamma \max_{a'}(Q(s',a')) - Q(s,a))$$

The Q-Learning has an exploration step in addition to the update step list above. However, since data already existed, the exploration step was replaced with iterating through the given data set. Updates were performed after each iteration of (s, a, r, s') tuple from the MIMIC-III data set of sepsis patients. The algorithm was implemented using $\alpha = 0.2$, $\gamma = 0.95$ for a total of 50 iterations.

IV. Experiment

Patient trajectories were derived using AI models and compared to those trajectories from physicians in the ICU using the MIMIC-III data set [7], which contains about 5,000 patients that meet Sepsis diagnostic criteria. We used the same patient cohort previously curated in [2]. The action space consisted of dosages of a vasopressor, which range from 0mg to 20mg and binned into 5 different ranges, and administration of IV fluids. The transition matrix $T(s'|s, a)$ for $s \rightarrow s'$ using patient features, chart events, and date-time events.

AI models were created and compared against physician behavior in a meaningful and intuitive manner. The dosage of both vasopressor drugs as well as IV fluids from AI models were compared to how physicians responded and treated patients. This assumes that the physicians' behavior of treatment approximated Bayesian error rates for treating sepsis patients. Therefore, models were analyzed to determine the extent to which they could achieve physician performance. In addition, the feature space was explored to determine whether projecting into non-linear space using VAE would provided better feature spaces for grouping patients into intuitive clusters. The AI models were evaluated using a Chi Squared test to determine the extent to which the policies derived mimicked the dosage for vasopressor and IV fluids that were administered from the physicians.

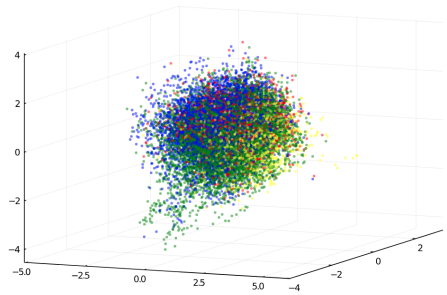

Fig. 1. Patient state space demonstrates an inability to clearly cluster patients into intuitive groupings

A. Patient Condition Space Clustering

Space clustering is one of the most critical parts of the problem. The challenges to cluster our high dimensional space are that: (a) we need to capture patient physiologic dynamics, (b) clustering should not mix patients who need different treatment plan to survive. We investigated kmeans++ clustering, used in Komorowski et al. [1], and a novel method of clustering using a VAE.

We observed that either clustering method failed to from well separated clusters. Figure 1 illustrates our challenges. Patients appear to aggregate into a single blob, regardless of survival status or MAP.

This issue motivated us to explore not only discrete space setting, but also continuous state space with value function approximation. Furthermore, for continuous space we joined previous states to capture patient condition dynamics and therefore improved the method.

B. Q-Learning Policies

The Q-learning algorithm provided an interesting policy for both deceased and survived patients as shown in Figures 2, 3 and 4. The figures demonstrate that the Q-Learning derived policies tend to be much more aggressive in dealing with sepsis patients than physicians, both in administering vasopressors as well as fluids.

However, it is quite interesting that the Q-Learning policy tends to be much more stringent when administering vasopressors as compared to fluids. In addition, the Q-Learning policy tends to be more conservative when administering vasopressors to patients that survived as opposed to those that expired.

Mean Arterial Pressure (MAP), is the single most important indicators of adequate systemic blood flow. We include it in figures 3 and 4 to compare clinician and AI policies in terms of a single variable. In the figures, it is clear that clinicians administers more vassopressors than the Q-learning policies, especially among deceased patients.

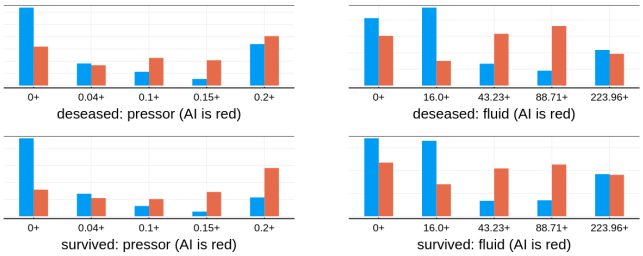

Fig. 2. Q-Learning policy for different sepsis states demonstrates a tendency to be much more aggressive in its administration of vasopressor and fluids.

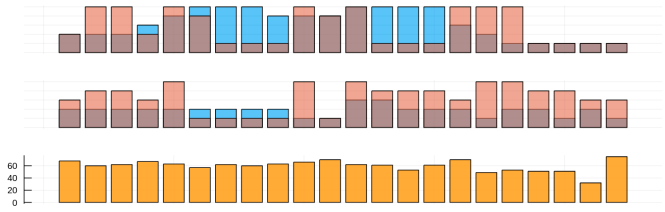

Fig. 3. Q-Learning. Discrete state space. Deceased patient: first row: vasopressor dosage, second: IV fluids, third: Mean Arterial Pressure. Clinician is in blue. AI is in red.

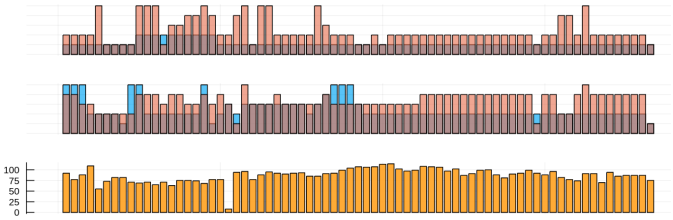

Fig. 4. Q-Learning. Discrete state space. Survived patient: first row: vasopressor dosage, second: IV fluids, third: Mean Arterial Pressure. Clinician is in blue. AI is in red.

C. Policy Iterator Policies

The policy iterator also defined a policy as shown in Figure 5. We see the same patterns of more aggressive use of vasopressors and IV fluids my the policy found through Policy Iteration when compared to clinician policies.

The Policy Iteration method showed an uncanny similarity to the Q-Learning method. This could be a result of the defined discrete space and action providing only a limited amount of information to derive the policies, in which case both methods identified a similar policy.

D. SARSA with Value Function Approximation

The SARSA method with VFA provides a notable departure from the previous discrete state and action methods. This continuous function method does not rely on the feature space clustering to generate a tractable state space. Instead, using a neural network, a non-linear transformation was accomplished directly on the feature space to transform it into a state space.

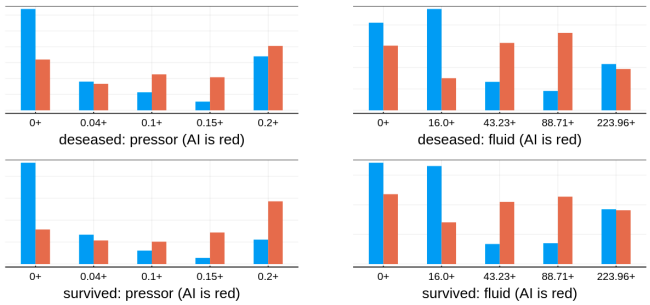

Fig. 5. Policy Iterator policy for different sepsis states demonstrates a tendency to be much more aggressive in its administration of vasopressor and fluids, similar to the Q-Learning algorithm.

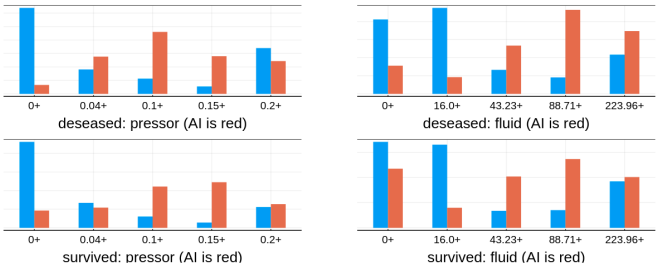

Fig. 6. SARSA with Value Function Approximation also tends to be more aggressive for administration of drugs, but more conservative when administering the highest doses of vasopressor compared to the other AI models' policies.

The algorithm provided much more impressive results compared to its discrete counterparts as shown in Figure 6. The vasopressor policy was much more aligned with clinicians' behaviors at highest dosage levels. This is reassuring as vasopressors, particularly in high doses, can cause other end organ dysfunction or even death. Nevertheless, we find the model's policies are still relatively more aggressive with respect to the administration of both vasopressors and IV fluids.

V. Discussion

This task required careful thought on how to manipulate notoriously messy electronic health records into a faithful representation of a patients physiological state and response to treatment as well as careful modeling decisions that balance computational complexity and the true nature of clinical practice.

We see these challenges come to bear as we could not find a reasonable well separated discrete state representation with enough nuance to capture the dynamics of sepsis. While Sepsis has a very high rate of mortality, the majority of patients will recover. Because there is no clear way to decide *a priori* which patients are likely to survive, researchers have no practical way of balancing these classes of patients in a given data set.

TABLE I
COMPARISON OF DOSAGE ADMINISTRATION TO PHYSICIAN POLICIES USING CHI SQUARED TEST

Power Divergence Results						
	SARSA		Q-learning		Policy Iterator	
Outcome and Fluid Type	Statistic	p-value	Statistic	p-value	Statistic	p-value
Deceased (IV Fluid)	22276.087	< 0.001	5076.620	< 0.001	13266.478	< 0.001
Survived (IV Fluid)	21845.965	< 0.001	7199.317	< 0.001	15858.067	< 0.001
Deceased (Vasopressor)	25989.214	< 0.001	9742.586	< 0.001	7263.196	< 0.001
Survived (Vasopressor)	60437.478	< 0.001	5131.762	< 0.001	4631.766	< 0.001

This help explains why a no-drug policy might be learned: because most patients are likely and require less treatment. This phenomena is evident in nearly identical policies found through Policy Iteration and Q-learning experiments that learn the same policies regardless of patient survival status. We were able to address this limitation through the application of SARSA with VFA, which allowed us to us a continuous state space thereby circumventing hard-coded modeling decisions. This method allowed our model to learn different policies for patients based on survival status without direct observation. Nevertheless, has shown in Table I, the AI models substantially deviated from physician policies of administering both IV fluids and vasopressors to patients whose outcome was both positive and negative with all below the $\alpha = 0.001$ threshold, rejecting the hypothesis that the policies are dependent or related. It should be worth noting that although the statistics measurement were extremely large, the Policy Iterator method had the smallest when dealing with vasopressors and the Q-Learning method had the smallest when dealing with IV fluids but produced a better overall fit. However, as previously discussed, they still aggressively administered vasopressors at the highest dosage as compared to SARSA with continuous states, a flaw that can potentially be life threatening.

While we show some improvement to modeling Sepsis treatment with RL by using a continuous state representation, there are significant limitations to our approach. In particular we did not address challenges with discrete 1-hour time windows. Response to rapid decompensation is perhaps the most challenging scenario for clinicians and the most difficult to research. The 1-hour time windows is, in all likelihood, much to large to capture such a phenomena and all its heterogeneity.

Overall we demonstrated ample opportunity for research in RL algorithms for clinician decision support. In particular, alternatives to discrete, fixed duration time windows are likely a vibrant area of research. Additionally, we see ample opportunity for more transparent external validation. While the MIMIC-III is a very large data set it only captures patients from two well connected hospitals in one area of the United States. Future research would benefit greatly from multi-center model evaluation. Finally, clinician input is vital to creating an algorithm that can truly be used in practice: close partnership with clinical domain experiments will help produce a algorithm that is both useful and trusted by medical experts.

VI. CONCLUSIONS

In this project, we applied several model based and model free RL algorithm to sepsis treatment dosing problem. We studied and compared discrete state model based and model free approaches and also continuous space SARSA algorithm with Neural Network VFA.

We were able to replicate the findings of the original work and address some its limitations with alternative state space representations and methods for solving for an optimal policy. In addressing the limitations, were were able to show application of SARSA with VFA better approximated the clinician policies and that Q-Learning was relatively indistinguishable from Policy Iteration. We observed that feature set, data size, and the nature of the algorithms all contribute to the overall performance of the algorithms. Our best models improved previous work by capturing more of patient state dynamics.

VII. CONTRIBUTIONS

All authors equally contributed to this project. Andriy focused on more dataset processing, state clustering, value iterator in discrete state setting, and Neural Network based SARSA algorithm. Robert focused on project paper, implementing discrete state algorithms such as Policy Iterator and Q-Learning as well as dataset processing and evaluation of policies. Sergio focused more on result analysis, project paper and discrete state algorithms.

Source code for the project is available at: https://github.com/rptrevin/AA228-Project.git

REFERENCES

[1] M. Komorowski, L. A. Celi, O. Badawi, A. C. Gordon, and A. A. Faisal, "The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care," vol. 24, no. 11, pp. 1716–1720. Number: 11 Publisher: Nature Publishing Group.

[2] R. Jeter, C. Josef, S. Shashikumar, and S. Nemati, "Does the "artificial intelligence clinician" learn optimal treatment strategies for sepsis in intensive care?,"

[3] D. A. Kaji, J. R. Zech, J. S. Kim, S. K. Cho, N. S. Dangayach, A. B. Costa, and E. K. Oermann, "An attention based deep learning model of clinical events in the intensive care unit," vol. 14, no. 2, p. e0211057. Publisher: Public Library of Science.

[4] S. Purushotham, C. Meng, Z. Che, and Y. Liu, "Benchmarking deep learning models on large healthcare datasets," vol. 83, pp. 112–134.

[5] A. Raghu, M. Komorowski, I. Ahmed, L. Celi, P. Szolovits, and M. Ghassemi, "Deep reinforcement learning for sepsis treatment,"

[6] M. Sotoodeh and J. C. Ho, "Improving length of stay prediction using a hidden markov model," vol. 2019, pp. 425–434.

[7] A. E. W. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database," vol. 3, no. 1, p. 160035. Number: 1 Publisher: Nature Publishing Group.

[8] M. Kochenderfer, T. Wheeler, and K. Wray, *Algorithms for Decision Making*. Stanford University, 2020.