Reinforcement Learning Project

1. Background

Gym-Sepsis (see GitHub gym-sepsis) is a reinforcement learning (RL) simulation environment, which is a part of the OpenAI Gym environments. It is built to simulate sepsis treatment in the intensive care unit (ICU). The state model, episode termination, and outcome model are trained on the MIMIC-III dataset of more than $40,000$ ICU patients. More details about the MIMIC-III dataset can be found here, and details about models can be found here. **The objective of this project is to learn the *optimal but also interpretable* treatment strategy for septic patients.**

Gym-Sepsis can be used to evaluate and train both offline policies and online policies. For example,

- Offline policies. To mimic the MIMIC-III data, one could use Gym-Sepsis to generate offline data and search for the optimal policy based on it.
- Online policies. Alternatively, one could consider running trials using Gym-Sepsis while searching for the optimal policy.

2. Environment Components

- For each time step, we have a 48×1 feature vector (46 physiological parameters as state, 1 treatment as action, and 1 time step index). Each time step corresponds to a **4-hour window** in the ICU, and an episode spans the entire ICU stay of a patient, with the time horizon determined by the length of the trajectory until discharge or death.
- The **state** was represented as a 46-dimensional vector of physiological features, including demographics, laboratory values, vital signs, and intake/output events. The full list of features, following the order in the implementation, is given below.
  - Feature names: albumin, anion gap, band neutrophils, bicarbonate, bilirubin, blood urea nitrogen (BUN), chloride, creatinine, diastolic blood pressure, glucose (two sources), heart rate, hematocrit, hemoglobin, international normalized ratio (INR), lactate, mean arterial blood pressure, arterial carbon dioxide pressure ($PaCO_2$), platelet count, potassium, prothrombin time (PT), partial thromboplastin time (PTT), respiratory rate, sodium, peripheral oxygen saturation (SpO_2), systolic blood pressure, body temperature in Celsius, white blood cell count (WBC), patient age, gender indicator, race indicators (white, Black, Hispanic, other), height, weight, ventilation status, SOFA score, LODS score, SIRS score, qSOFA score, qSOFA sub-scores for systolic blood pressure, Glasgow Coma Scale, and respiratory rate, Elixhauser comorbidity index, and blood culture positivity. Further details about the feature names and units can be found here.
  - The feature names are also introduced on line 21 in `sepsis_env.py`
- The **action** space was defined by a discrete 5×5 grid over medical interventions, spanning intravenous (IV) fluid and maximum vasopressor (VP) dosage within a 4-hour window. Each drug was discretized into quartile bins based on all non-zero dosages, with an additional bin 0 representing no drugs. Thus, at each timestep, the treatment choice was represented as a vector of (total IV input, maximum VP input), encoded as an integer by flattening the two-dimensional grid. Specifically,

$$\text{action} = 5 \times \text{IV\_bin} + \text{VP\_bin}, \quad \text{IV\_bin} \in \{0, \ldots, 4\}, \quad \text{VP\_bin} \in \{0, \ldots, 4\}$$

.

- The **immediate reward** was defined as $+15$ at the terminal step if the patient was discharged from the hospital and -15 if the episode ended in death, while all intermediate steps received immediate reward 0.

  You are allowed to adopt a different choice of the immediate reward, such as the one described in Section 7.4 of [1]. You may modify line 82 in `sepsis_env.py` by changing `reward = 0` to another function.

3. Implementation

Gym-Sepsis was originally built between 2018 and 2019. Its many required package versions are no longer compatible with the latest Python releases. Therefore, please ***make sure to build a conda or mamba environment with Python 3.6*** before using Gym-Sepsis.

Alternatively, you may use a modified version of Gym-Sepsis which is compatible with Python 3.10.

Gym-Sepsis has been registered in OpenAI's Gym interface, which allows it to be used with RL libraries such as MushroomRL and d3rlpy that incorporate several state-of-the-art RL algorithms in Python. Please refer to the examples in MushroomRL or d3rlpy to explore how RL algorithms can be integrated with the Gym-Sepsis environment. One should first verify that the installed library version matches the example code; if not, one needs to consult the example code corresponding to the installed version.

4. Project Report and Submission

A project report should be written by a group of 1-3 people. In the report, each group should:

- describe and specify the mathematical formulation of the problem;
- describe and apply an approach (existing or newly developed);
- analyze, interpret, and discuss the results;
- evaluate the performance of the approach according to a set of metrics, preferably against a baseline approach.

The report should be prepared following Instructions for authors for the *Journal of the American Statistical Association*, where both Word and LaTeX templates are also provided. **The report should be no more than 25 pages. It does not need to be anonymized.** *When working in groups, please clearly detail the contributions of individual group members.*

Project Submission. The following files should be **emailed to the instructor by Oct 27th, 11:59 PM, EST.**:

- Project Report (Required): a single PDF file with no more than 25 pages.
- Code File (Required): either a single file or a zip file is allowed.
- Supplementary Document (Optional): a single PDF file but it should not contain any essential components for readers to understand the project report; the project report itself should be self-contained.

Multiple submissions are allowed but only the last submission before the deadline will be graded.

All projects will be graded with the same standard regardless of the number of people in the group. The final grade will be based on the following criteria:

- Structure of the paper

- Writing quality
- Appropriate use and evaluation of the approach
- Clarity of figures and tables
- Appropriateness of abstract and keywords
- Quality of discussion and conclusions
- Adequacy of references

References

[1] A. Raghu, M. Komorowski, I. Ahmed, L. Celi, P. Szolovits, and M. Ghassemi. Deep reinforcement learning for sepsis treatment. *arXiv preprint arXiv:1711.09602*, 2017.