Harvard John A. Paulson School of Engineering and Applied Sciences

Evaluation of Optimal Decision Making with Dead-ends in High-Risk Environments

Alex Arron Kashi and Marcel Torne Villasevil

Advised By
Professor Susan Murphy PhD., Harvard
Raaz Dwivedi PhD., Harvard
Eura Shin, Harvard

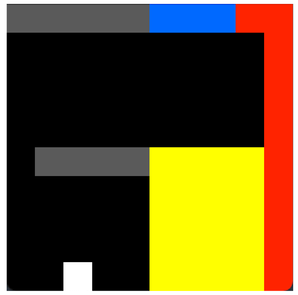
Motivation

Dead-ends are **states** that will certainly **lead** the agent to a **negative terminal state**. Dead-end identification is a significant problem that occurs in a variety of fields such as **healthcare** (with identifying these in treatments), **robotics** (identifying crashes before these happen), **stock markets** (identifying when the stocks are just going to lose value), and many more. Finally, it is a non-trivial problem, and in this project, we study it further.

Environment

Adapted from Fatemi et al.[1].

- Terminal State - Death
- Goal State - Cure
- Dead-end - Terminally ill
- Wall - Barrier to Treatment
- Field - Neutral
- Agent - Patient



State Space

Agent position (y, x)

Action Space

No-op, Up, Down, Left, Right

Transitions

Death drag - 20% chance of a forceful right

In dead-end? - 70% chance of a forceful right, otherwise no-op

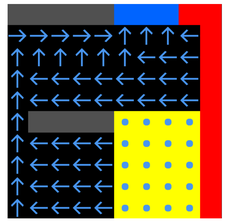
Rewards

Goal state (1)
Terminal State (-1)

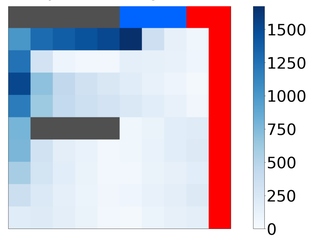
Objective

1. Identify dead-ends (propose different method)
2. Evaluate how the data collection policy affects each method
3. Learn a policy that can drive agents to recovery states

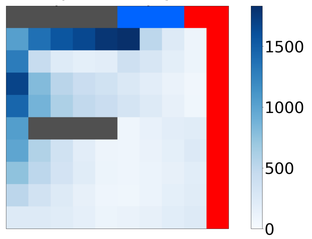
Optimal Policy



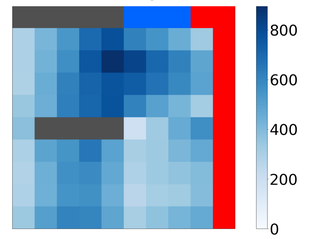
Optimal Trajectories



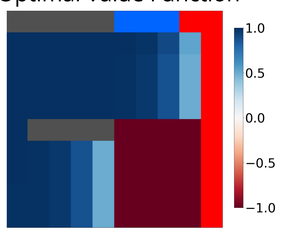
Suboptimal Trajectories



Random Trajectories



Optimal Value Function



$$V^*(s) = \max_a Q^\pi(s,a)$$

Vary the way of **collecting data** for posterior Offline RL:

1. Using an **optimal policy** (Policy obtained through value iteration and policy extraction)
2. Using a **suboptimal policy** (with 0.25 probability, it will take random action; otherwise, it takes the optimal action)
3. Using a **random policy** (random actions at each timestep)

Learn **Value function** using different algorithms:

1. **Q-Learning**[3].
2. Dead-end discovery method by **Fatemi** et al[1].
3. **Fitted Q Iteration (Deep Learning)**

Methods

Algorithm 14 Fitted Q iteration

procedure FQI
  Initialize the neural network weights ϕ randomly
  while not terminate **do**
    Collect dataset $\mathcal{D}$ using an arbitrary policy to interact with the environment
    for $i = 1 \cdots$ **do**
      Compute the target $y \leftarrow r(s,a) + \gamma \max_{a'} Q_\phi(s',a') \; \forall (s,a,r,s') \in \mathcal{D}$
      Update the network $\phi \leftarrow \arg\min_\psi \frac{1}{|\mathcal{D}|} \sum_{(s,a,r,s') \in \mathcal{D}} \|y - Q_\psi(s,a)\|^2$
    end for
  end while
end procedure

Algorithm 12 Value Iteration

procedure ValueIteration(π, ϵ)
  Initialize $V_1(s) = 0 \; \forall s \in \mathcal{S}$
  for $i = 1 \cdots$ **do**
    $V_{i+1}(s) \leftarrow \max_a \sum_{s' \in \mathcal{S}} \mathcal{T}(s'|s,a)\left[r(s,a) + \gamma V_i(s')\right] \; \forall s \in \mathcal{S}$
    if $\|V_{i+1} - V_i\|_\infty \leq \epsilon$ **then**
      break
    end if
  end for
  Set policy $\pi(s) = \arg\max_a \mathbb{E}_\pi\left[r(s_t,a_t) + \gamma V^\pi(s_{t+1}) | s_t = s, a_t = a\right] \; \forall s \in \mathcal{S}$
end procedure

Algorithms referenced from MIT 6.484 lecture notes[2].

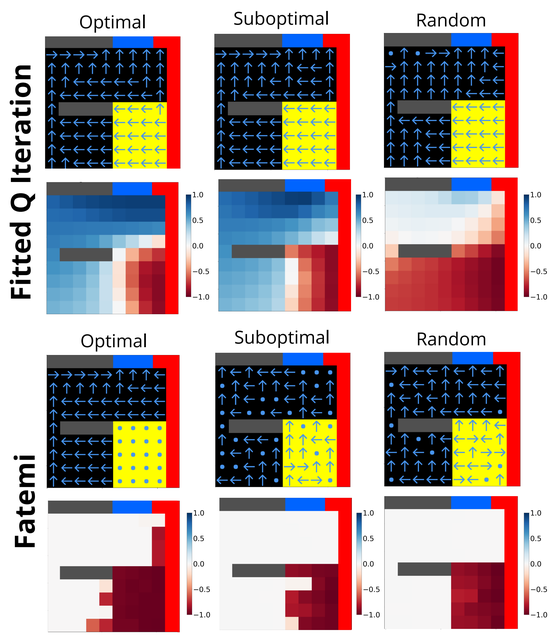
Results

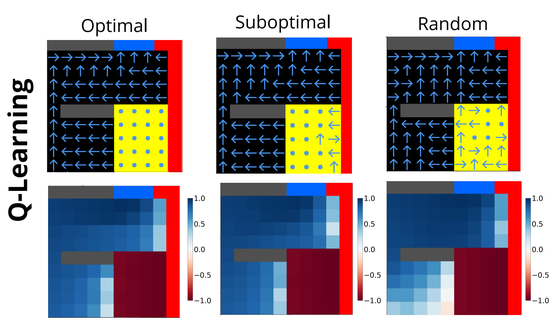
Experimental Setup:

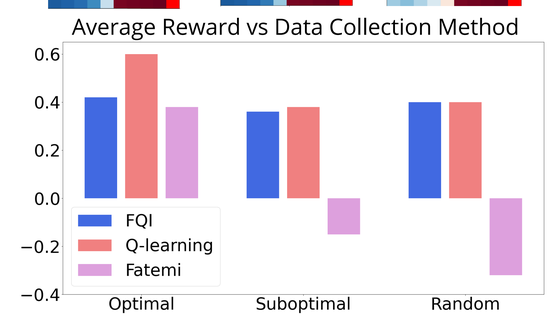
- Collect **2500** trajectories
- Max length of **100** steps.

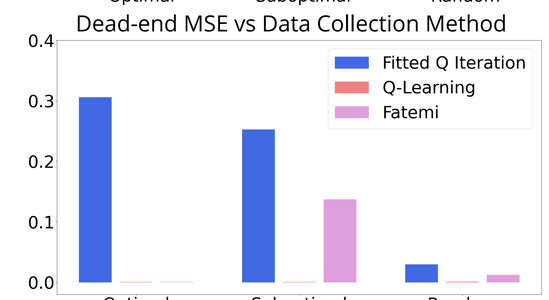
Analysis:

- Fatemi's method produces results that are worse than Q-learning. We can deduce that knowing which states lead to recovery gives valuable information about dead ends.
- Regarding Q-Learning, no matter how the data was generated, it still recovers a reasonable estimation of the value function.
- Regarding Fitted Q Iteration, this is not the case. The best approximation is found for the data collected using the optimal policy.
- Concerning Fatemi's method, all recover a proper value function, but the best is found for the random data.









Conclusion

All methods studied in this project, Fitted Q Iteration, Q-Learning, and the method proposed by Fatemi et al., all **manage to identify** the **dead-ends** in the proposed toy environment. Furthermore, we saw that with the different ways that the data could be collected, **most** of these **methods** managed to **recover the Q-values.** Nevertheless, we observe that the results are the **most consistent** for **Q-Learning.** In addition, this method needs less training time than Fitted Q Iteration, and for this reason, given out experiments, **we would propose Q-Learning for the dead-end identification.**

References

1. Fatemi, Mehd0i, et al. "Medical Dead-ends and Learning to Identify High-risk States and Treatments." Advances in Neural Information Processing Systems 34 (2021).
2. Zhang-Wei Hong, Pulkit Agrawal, Lecture Notes of 6.484 Computational Sensorimotor Learning, MIT, 2022
3. Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.