Paper Critique

Shuvrajeet Das, DA24D402

Course: DA7400, Fall 2024, IITM
Paper: [Best-Action Imitation Learning (BAIL) (Curse you, bayle!!)]
Date: [09-08-2024]

Make sure your critique Address these following points:
1. The problem the paper is trying to address
2. Key contributions of the paper
3. Proposed algorithm/framework
4. How the proposed algorithm addressed the described problem
Note: Be concise with your explanations. Unnecessary verbosity will be penalized. Please don't exceed 2 pages.

Offline Reinforcement learning where the main aim is to get a very high rewarding policy based on the dataset that is created using a behavioural policy π_β, the behavioural policy could be anything like from a random policy to a highly sophistic heuristic based approach policy etc. This gives us a historical interactions based dataset where we train our model.

Traditional Q-function-based algorithms, like Deep Deterministic Policy Gradient (DDPG), often fail in this context due to issues like extrapolation error, which leads to a poor and diverging performance. Therefore, the paper proposes a new algorithm, Best-Action Imitation Learning (BAIL), which is designed to achieve high performance while being both conceptually and computationally simpler than existing approaches. The goal of this approach is to improve policy learning by effectively leveraging the data available in the batch, providing stable value estimates, and avoiding the pitfalls of conventional methods.

The paper mainly focuses on the imitation learning of the best actions that is taken in the dataset. This makes it able to learn the policy in a simple, effective and a stable way without any sort of interaction with the environment.

Algorithm 1 BAIL

Initialize upper envelope parameters ϕ, ϕ', policy parameters θ. Obtain batch data $\mathbb{B}$. Randomly split data into training set $\mathbb{B}_t$ and validation set $\mathbb{B}_v$ for the upper envelope.
Compute return G_i for each data point i in $\mathbb{B}$.
Obtain upper envelope by minimizing the loss $L^k(\phi)$:
for $j = 1, \ldots, J$ **do**
    Sample a mini-batch $\mathbb{B}$ from $\mathbb{B}$.
    Update ϕ using the gradient: $\nabla_\phi \sum_{i \in \mathbb{B}} (V_\phi(s_i) - G_i)^2 \Big[\mathbf{1}\{(V_\phi(s_i) > G_i)\} + K\mathbf{1}\{(V_\phi(s_i) < G_i)\} \Big] + \lambda ||\phi||^2$
    if time to do validation for the upper envelope **then**
        Compute validation loss on $\mathbb{B}_v$
        Update ϕ and ϕ' according to the validation loss
    end if
end for
Select data point i if $G_i > zV_\phi(s_i)$, where z is such that $p\%$ of data in $\mathbb{B}$ are selected. Let $\mathbb{U}$ be the set of selected data points.
for $l = 1, \ldots, L$ **do**
    Sample a mini-batch $\mathbb{U}$ of data from $\mathbb{U}$.
    Update θ using the gradient: $\nabla_\theta \sum_{i \in \mathbb{U}} (\pi_\theta(s_i) - a_i)^2$
end for

BAIL is designed to be conceptually straightforward and computationally less intensive compared to existing batch reinforcement learning methods. It avoids complex operations like Q-learning, which often suffer from extrapolation errors in the batch setting. The way tries to address the problem can be described in 3 step:

- **Penalty Loss Function** $L(\phi)$ The penalty loss function $L(\phi)$ is a crucial component in the BAIL algorithm that is used to learn the upper envelope $V_\phi(s)$ of the value function. The upper envelope function $V_\phi(s)$ is an estimate that aims to be close to the true returns G_i for each state s in the dataset, which is also regularized by adding some constraints.

$$L^k(\phi) = \sum_{i \in \mathbb{B}} \left(V_\phi(s_i) - G_i\right)^2 \left[\mathbf{1}\{V_\phi(s_i) > G_i\} + K\mathbf{1}\{V_\phi(s_i) < G_i\}\right] + \lambda||\phi||^2$$

- **Hyperparameter p for Selecting Data** The hyperparameter p plays a vital role in selecting the top percentage of data points from the batch dataset $\mathbb{B}$. Specifically, after the upper envelope $V_\phi(s)$ is learned, the BAIL algorithm selects a subset of data points where the return G_i is greater than a scaled version of the estimated upper envelope $z \cdot V_\phi(s_i)$. The hyperparameter p determines the threshold z such that only the top $p\%$ of data points with the highest advantage (i.e., the highest likelihood of being optimal actions) are selected.

- **BAIL = Upper Envelope + Imitation Learning** The BAIL framework is basically an amalgamation of two main components: Upper Envelope Estimation and Imitation Learning. These two components work together to enable the BAIL algorithm to effectively learn a high-performing policy from a fixed batch of data.