Deep Reinforcement Learning (Sp25)

Instructor: Dr. Mohammad Hossein Rohban

Summary of: Lecture 25

Summarized By: Behnia Soleymani

- **1. Implicit Q-Learning (IQL)**
  - **Goal**: Avoid OOD actions in the Q-learning target $r + \gamma \max_{a'} Q(s', a')$ by not explicitly using $\max_{a'}$ over the learned policy.
  - **Key Idea**:
    (a) Learn a state-value function $V(s)$ that approximates the value of the best actions seen in the dataset for state s. This is done using expectile regression:

$$V \leftarrow \arg\min_V \sum_i l_\tau^2(V(s_i), Q(s_i, a_i))$$

    where $(s_i, a_i) \sim \mathcal{D}$ (from the dataset) and $l_\tau^2(u) = |\tau - \mathbb{I}(u < 0)|u^2$ is the expectile loss. A large τ (e.g., 0.7-0.9) pushes $V(s_i)$ towards the higher end of observed $Q(s_i, a_i)$ values.

    (b) Update the Q-function using this $V(s')$ as the target for the next state's value:

$$Q(s, a) \leftarrow r(s, a) + \gamma V(s')$$

    (c) Extract the policy $\pi(a|s)$ by, for example, Advantage-Weighted Regression (AWR), using advantages $A(s, a) = Q(s, a) - V(s)$. This implicitly encourages actions with high Q values relative to the dataset's average goodness $V(s)$.
  - **Benefit**: Q-function updates are grounded in values derived from dataset actions, mitigating OOD issues in targets.
- **2. Conservative Q-Learning (CQL)**
  - **Goal**: Directly penalize overestimated Q-values for OOD actions while fitting Q-values to the dataset.
  - **Key Idea**: Modify the standard Bellman error objective with a conservative regularizer. The CQL loss $L_{CQL}(Q)$ is:

$$\begin{aligned} L_{CQL}(Q) = \alpha \big(&\mathbb{E}_{s\sim\mathcal{D}, a\sim\mu(a|s)}[Q(s, a)] - \mathbb{E}_{(s,a)\sim\mathcal{D}}[Q(s, a)]\big) \\ &+ \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[(Q(s, a) - (r(s, a) + \gamma \mathbb{E}_{a'\sim\pi}[Q(s', a')]))^2\right] \end{aligned}$$

  - **Term 1 (Regularizer)**: $\mathbb{E}_{s\sim\mathcal{D}, a\sim\mu(a|s)}[Q(s, a)]$: Pushes down Q-values for actions sampled from a distribution μ (e.g., current policy, or actions sampled to maximize Q). This targets potentially OOD actions with high Q.
    $\mathbb{E}_{(s,a)\sim\mathcal{D}}[Q(s, a)]$: Pushes up Q-values for actions actually present in the dataset $\mathcal{D}$.
    α: Hyperparameter controlling the strength of this regularization.
  - **Term 2 (Bellman Error)**: Standard TD error for the current policy π.
  - **Policy Update**: Update policy π based on the learned conservative $\hat{Q}$ (e.g., using SAC-style actor update or AWR).

- **Guarantee (Informal)**: For large enough α, the expected value under π of the learned $\hat{Q}$ is a lower bound on the true value of π, for states in $\mathcal{D}$.
- **Benefit**: Provides a more direct mechanism to prevent Q-value overestimation.

- **3. Model-Based Offline RL**
  - **General Idea**: Learn a dynamics model $\hat{p}(s'|s,a)$ from $\mathcal{D}$, then use $\hat{p}$ for policy learning (e.g., planning, generating synthetic data).
  - **Challenge**: Model Exploitation. The policy might learn to exploit inaccuracies in $\hat{p}$, especially in OOD regions not well-covered by $\mathcal{D}$, leading to compounding errors during model rollouts.
  - **Key Algorithms**:
    * **MOPO (Model-Based Offline Policy Optimization)**: **Idea**: Penalize the policy for entering regions where the model $\hat{p}$ is uncertain. **Mechanism**: Modify rewards used for planning/learning with the model:

      $$\tilde{r}(s,a) = r(s,a) - \lambda u(s,a)$$

      where $u(s,a)$ is an estimate of the model's uncertainty (e.g., disagreement among an ensemble of models) for the transition (s,a). λ controls penalty strength.
    * **COMBO (Conservative Offline Model-Based Policy Optimization)**: **Idea**: Apply CQL-like conservatism to Q-values learned with the model. **Mechanism**: Similar to CQL, add a regularizer that minimizes Q-values for state-action pairs (s_m, a_m) generated by rolling out the learned model $\hat{p}$, while maximizing Q-values for real data pairs (s_d, a_d) from $\mathcal{D}$.

      $$\min_Q \beta \left(\mathbb{E}_{(s,a)\sim\rho_{\text{model}}}[Q(s,a)] - \mathbb{E}_{(s,a)\sim\mathcal{D}}[Q(s,a)] \right) + \text{BellmanError}_{\text{model}}(Q)$$

      (Bellman error is computed using the learned model $\hat{p}$).

- **4. Which Offline RL Algorithm to Use?**
  - **Purely Offline Training**:
    * CQL: Well-understood, widely tested, fewer hyperparameters (mainly α).
    * IQL: More flexible, can also be good for finetuning.
  - **Offline Training + Online Finetuning**:
    * AWAC (Advantage-Weighted Actor-Critic): Implicit policy constraint method, widely used.
    * IQL: Often shows strong performance in finetuning.
  - **If a Good Dynamics Model Can Be Trained**:
    * COMBO/MOPO: Can leverage model benefits, but model quality is critical.

- **Multi-Agent RL and Game Theory**
  - **1.1 The Challenge of Multi-Agent Reinforcement Learning (MARL)**:
    - Unlike single-agent Reinforcement Learning (RL) where an agent learns in a (typically assumed) stationary environment, Multi-Agent Reinforcement Learning (MARL) involves multiple adaptive agents learning simultaneously. The core challenge arises because from any single agent's perspective, the environment becomes non-stationary. As other agents learn and change their policies, the transition dynamics and reward structures effectively change over time. Thus, each agent must explicitly account for other agents' actions and behaviors to make effective decisions.
  - **1.2 Game Theory: A Framework for Strategic Interaction**:
    - Game Theory provides the mathematical tools to model and analyze these strategic interactions. It formally studies scenarios involving multiple decision-makers, termed "players" or "agents," who are assumed to be rational (choosing actions to best achieve their objectives) and self-interested (primarily caring about their own benefits or payoffs). Critically, the outcome for each agent depends not only on its own actions but also fundamentally on the actions of all other agents involved in the game.
  - **1.3 How Game Theory Shapes MARL**:
    - Game Theory offers a principled foundation for MARL by defining what constitutes rational decision-making in a multi-agent context where payoffs are interdependent. It provides models, such as Normal Form Games, to represent these complex interactions and identifies "solution concepts" like Dominated Strategies and Nash Equilibria. These solution concepts can be interpreted as stable points or target behaviors that MARL agents might learn to converge towards, or they can guide the design of MARL algorithms by providing a theoretical understanding of desirable collective outcomes.
  - **1.4 Normal Form Games: A Basic Model of Interaction**:
    - A fundamental representation within Game Theory is the Normal Form Game (or strategic form game), which models a one-shot interaction. It consists of a set of two or more agents, each possessing a set of available actions. Agents are presumed to choose their actions simultaneously, and a specific combination of one action from each agent forms a "strategy profile" or "joint action." The consequence of this joint action is defined by a reward function for each agent, mapping the joint action to a scalar reward, without an explicit notion of evolving states within this single interaction.
- **2. Predicting Behavior with Game Theoretic Solution Concepts**
  - **Strictly Dominated Strategies**:
    - **Definition**: A strategy a_i for agent i is strictly dominated if there exists another strategy a'_i for agent i such that a'_i yields a strictly higher payoff than a_i for all possible combinations of actions by the other players (a_{-i}). Formally: $\exists a'_i, \forall a_{-i}, R_i(a_i, a_{-i}) < R_i(a'_i, a_{-i})$.

* **Implication**: A rational agent will never play a strictly dominated strategy. This allows for game simplification through Iterated Elimination of Strictly Dominated Strategies (IESDS).
* The Prisoner's Dilemma is a classic example where IESDS leads to a unique outcome: (Confess, Confess).

– **Nash Equilibrium (NE)**:

* **Motivation**: Strict dominance doesn't always identify a unique outcome, or even simplify the game at all.
* **Definition**: A strategy profile $a^* = (a_1^*, \ldots, a_N^*)$ is a Nash Equilibrium if no agent has an incentive to unilaterally deviate from its chosen strategy a_i^*, given that all other agents a_{-i}^* stick to their strategies.
* Formally: $\forall i, \forall a_i' \in A_i, R_i(a_i^*, a_{-i}^*) \geq R_i(a_i', a_{-i}^*)$.
* Equivalently, in an NE, each agent's strategy a_i^* is a best response to the strategies a_{-i}^* being played by the other agents.