Paper Critique

Shuvrajeet Das, DA24D402

Course: DA7400, Fall 2024, IITM
Paper: [Explainable Reinforcement Learning via Reward Decomposition]
Date: [01-11-2024]

Make sure your critique Address the 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.

---

1 The problem the paper is trying to address

The paper addresses the problem of making reinforcement learning (RL) decisions explainable by decomposing rewards into semantically meaningful components. Traditional Q-values in RL aggregate reward signals, making it difficult to discern the underlying motivations for specific actions. This work proposes a framework for reward decomposition that enables the separation of rewards into different types, facilitating action comparison based on trade-offs among these types.

The mathematical formulation for this problem is based on decomposing the reward function $R(s, a)$ in a Markov Decision Process (MDP) into a vector of reward types where $R_c(s, a)$ represents the reward for type c in state s and action a, and C is the set of all reward types.

2 Key contributions of the paper

- **Reward Decomposition Framework:** Introduction of a reward decomposition framework in reinforcement learning (RL) that separates the overall reward $R(s, a)$ into multiple meaningful reward types $R_c(s, a)$ for better interpretability. This decomposition allows the total Q-function $Q^\pi(s, a)$ to be expressed as:

$$Q^\pi(s, a) = \sum_{c \in C} Q_c^\pi(s, a)$$

  where $Q_c^\pi(s, a)$ represents the Q-value associated with each reward type c.

- **Minimal Sufficient Explanation (MSX):** Proposal of the minimal sufficient explanation (MSX) to explain why one action is preferred over another. For two actions a_1 and a_2 in state s, the reward difference explanation (RDX) is defined as:

$$\Delta(s, a_1, a_2) = \mathbf{Q}(s, a_1) - \mathbf{Q}(s, a_2)$$

  where $\mathbf{Q}(s, a)$ is a vector of decomposed Q-values. The MSX is a minimal subset of reward types that sufficiently explain the preference between a_1 and a_2.

- **Convergent Off-policy Algorithm (drQ):** Development of an off-policy decomposed reward Q-learning algorithm (drQ) that provably converges to an optimal policy and accurately estimates decomposed Q-values. This ensures that each Q_c for reward type c converges to the correct component Q-value, achieving:

$$Q^\pi(s, a) = \sum_{c \in C} Q_c^\pi(s, a)$$

3 Proposed algorithm/framework

Algorithm 1 Table-Based Decomposed Reward RL

$s_0 \leftarrow$ Initial State
$a_0 \leftarrow \epsilon(Q^0, s_0)$
$t \leftarrow 0$
repeat
  $(s_{t+1}, \mathbf{r}_t) \leftarrow \text{Act}(a_t)$
  $a_{t+1} \leftarrow \epsilon(Q^t, s_{t+1})$ ▷ ϵ-greedy exploration
  for all $c \in C$ **do**
    if drQ **then**
      $a' \leftarrow \arg\max_a \sum_c Q_c^t(s, a)$
    else if HRA **then**
      $a' \leftarrow \arg\max_a Q_c(s, a)$
    else if drSARSA **then**
      $a' \leftarrow a_{t+1}$
    end if
    $Q_c^{t+1}(s_t, a_t) \leftarrow (1 - \alpha) Q_c^t(s_t, a_t) + \alpha \left(r_{t,c} + \gamma Q_c^t(s_{t+1}, a')\right)$
  end for
  $t \leftarrow t + 1$
until convergence

4 How the proposed algorithm addressed the problem

- **Reward Decomposition**: Instead of aggregating all rewards into a single Q-value, the algorithm maintains separate Q-values $Q_c(s, a)$ for each reward type $c \in C$.
- **Action Selection Based on Decomposed Q-values**:
  - **drQ**: Chooses actions by maximizing the sum of all decomposed Q-values $\sum_c Q_c(s, a)$, making it transparent how different reward types influence the overall policy.
  - **HRA (Hierarchical Reward Architecture)**: Selects actions by focusing on a specific reward type, allowing analysis of actions under isolated reward considerations.
  - **drSARSA**: Uses the next action a_{t+1} as chosen by ϵ-greedy exploration, which supports continuous, on-policy updates and ensures consistency with decomposed learning.
- **Update Rule with Decomposed Rewards**: For each reward type c, the algorithm updates $Q_c(s, a)$ using the observed reward $r_{t,c}$ of that type, ensuring that each Q-value accurately reflects its associated component of the cumulative reward:

$$Q_c^{t+1}(s_t, a_t) \leftarrow (1 - \alpha) Q_c^t(s_t, a_t) + \alpha \left(r_{t,c} + \gamma Q_c^t(s_{t+1}, a')\right)$$

  This decomposition allows specific analysis of each reward type's influence on action choices, directly addressing the need for interpretable reinforcement learning.
- **Minimal Sufficient Explanation (MSX)**: By decomposing Q-values, the algorithm supports *Minimal Sufficient Explanations* (MSX) for why one action is preferred over another. The MSX highlights the minimum set of reward types that justify the preference, making it easier to understand agent behavior in complex environments.