Dots and Boxes RL: MDP Formulation and Q-Learning Analysis

Authors: Obidur Rahman (University of Chittagong, Bangladesh)
Date: 2025

Abstract

This repository formalizes the game of *Dots and Boxes* as a finite Markov Decision Process (MDP) and provides a rigorous convergence proof for tabular Q-learning. A proof-of-concept implementation for a 2x2 board is included, with placeholders for empirical results.

Project Overview

This project investigates the application of Reinforcement Learning (RL) to the combinatorial game *Dots and Boxes*. Key contributions include:

1. **MDP Modeling**: A formal mathematical framework for the game.
2. **State-Space Analysis**: Exact combinatorial bounds on possible game states.
3. **Q-Learning Convergence**: Proof of convergence under standard RL assumptions.
4. **Implementation**: Pseudocode and design for a 2x2 board experiment.

Key Contributions

1. Combinatorial State-Space Analysis

The state-space size grows exponentially with board size n:

- **Edges**: $E(n) = 2n(n+1)$ (horizontal + vertical edges).
- **State count**: $|S| \le 2^{E(n)} \times 2$ (edge subsets × current player).

Board Size (n)	Edges (E(n))	Total States (|S|)
1x1	4	32
2x2	12	8,192
3x3	24	~33.6 million
4x4	40	~2.2 trillion

---

2. MDP Formulation

The game is modeled as an **episodic MDP** $M = (S, A, P, R, \gamma)$:

- **States**: $s = (B, p)$ where B is a bitmask of drawn edges and $p \in \{1,2\}$ is the current player.
- **Actions**: Legal moves = undrawn edges in state s.
- **Transitions**:
  - If the agent completes a box, it moves again.
  - Otherwise, the opponent (random policy) plays.
- **Rewards**: $R(s,a)$ = number of boxes completed by the agent's move.

- **Discount**: $\gamma = 1$ (finite episode).

---

3. Q-Learning Convergence

Theorem 1: Under standard conditions (finite state/action spaces, stationary MDP, ε-greedy exploration, and decaying learning rates), tabular Q-learning converges to the optimal policy.

```
Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + α_t [r_{t+1} + γ max_{a'} Q_t(s_{t+1}, a') -
Q_t(s_t, a_t)]
```

---

4. 2x2 Board Implementation

Hyperparameters:

- Learning rate $\alpha = 0.1$
- Exploration rate $\varepsilon = 0.1$
- Episodes: N (placeholder).

State Encoding:

- 12-bit mask for edges + 1 bit for current player.

Pseudocode:

```
Initialize Q(s,a) = 0 for all states and actions.
for episode in 1..N:
    s = initial state
    while not terminal(s):
        a = ε-greedy action from Q(s,·)
        (s', r, done) = env.step(a)
        if r == 0:
            (s', r_opp, done) = env.opponent_step()
            r = r - r_opp
        Q(s,a) = Q(s,a) + α[r + max_{a'} Q(s',a') - Q(s,a)]
        s = s'
```

Future Work

- **Scaling**: Use function approximation (e.g., neural networks) for $n \geq 3$.
- **Hierarchical RL**: Exploit game structure for abstraction.
- **Empirical Validation**: Complete placeholder figures/tables for 2x2 results.

Citation

```
@article{rahman2025dots,
  title = {A Rigorous MDP Formulation and Q-Learning Convergence Analysis for Dots
and Boxes},
  author = {Rahman, Obidur},
  journal = {arXiv preprint arXiv:2507.XXXXX},
  year = {2025}
}
```