Introduction

Optimizing Decision-Making in Multi-Agent RL with CPT

- Investigating Multi-Agent Reinforcement Learning (MARL) under Cumulative Prospect Theory (CPT)
- Key motivation: **Aligning autonomous agents with human decision-making biases**
- Focus areas:
  - Utility and probability distortions in MARL
  - Strategy optimization in multi-agent interactions
  - Emergent behaviors in mixed agent populations
  - Strategic information elicitation

Background

Cumulative Prospect Theory (CPT) & MARL

- **Traditional RL**: Agents maximize expected rewards
- **CPT Agents**: Modify reward and probability perception
  - Reference-dependent evaluation
  - Loss aversion: More sensitive to losses than gains
  - Nonlinear value and probability weighting functions
- **MARL Setting**: Multi-agent interactions in cooperative, competitive, and mixed-motive environments

MARL Formulation

Mathematical Framework

- **Markov Decision Process (MDP)**:
  - States, actions, transition probabilities, rewards, discount factor
- **Multi-Agent Extension**:
  - Multiple agents optimizing individual rewards
  - Interaction through joint action space
  - Nash equilibrium as a classical solution concept
- **CPT Integration**:
  - Agents optimize for prospect-theoretic utilities rather than expected rewards

CPT-Driven Reinforcement Learning

How CPT Alters RL Decision-Making

- **Value Function**: Loss aversion and diminishing sensitivity

$$v(x) = \begin{cases} x^\alpha, & x \geq 0 \\ -\lambda(-x)^\alpha, & x < 0 \end{cases}$$

- **Probability Weighting**: Overweighting rare events, underweighting frequent events

$$w(p) = \frac{p^\beta}{(p^\beta + (1-p)^\beta)^{1/\beta}}$$

- **Policy Optimization Challenge**:
  - Nonconvexity in probability and value transformations
  - No Bellman equation, making dynamic programming ineffective

Implementation Strategy

Technical Approach & PyTorch Implementation

- **Policy Gradient Optimization with CPT**
  - CPT-adjusted rewards & probability distortions
  - Model-free learning using policy gradients
- **Implementation Workflow**:
  1. Design neural network for policy representation
  2. Transform rewards using CPT functions
  3. Compute policy gradients using automatic differentiation
  4. Optimize policies using gradient ascent
- **Evaluation**:
  - Multi-agent simulations (PettingZoo, Gym)

Initial Results

Current Progress & Observations

- **MARL Training with MADDPG Successfully Implemented**
    - Reward trends show learning progress
    - Policy updates and replay buffer working as expected
- **Challenges**:
    - No CPT mechanisms integrated yet
    - Stability issues in multi-agent coordination
    - Absence of explicit evaluation metrics (e.g., win/loss ratio, episodic scores)

Next Steps & Challenges

Planned Improvements

- **CPT Integration**
  - Implement probability weighting and value distortions
  - Modify policy updates for CPT-weighted objectives
- **Technical Hurdles**
  - Gradient stability under CPT-induced reward transformations
  - Multi-agent coordination under risk-sensitive behaviors
  - Computational overhead from probability-weighted updates
- **Open Questions**
  - How does CPT impact equilibrium stability?
  - Best strategies for CPT-weighted return approximation?

Conclusion

Summary & Future Directions

- MARL framework successfully implemented, but **CPT integration pending**
- Policy gradient approach chosen for adaptability to nonconvex objectives
- Early results validate **agent learning**, but evaluation metrics need refinement
- **Next Steps**:
  - Incorporate CPT-based distortions
  - Improve training stability & evaluation methods
  - Assess strategic behaviors under CPT in multi-agent environments

Thank You! Questions?