Reinforcement Learning in Game Theory

ECE 270

Matthew Blischke, Bryan Jang, Eddie Zhang

December 2, 2021

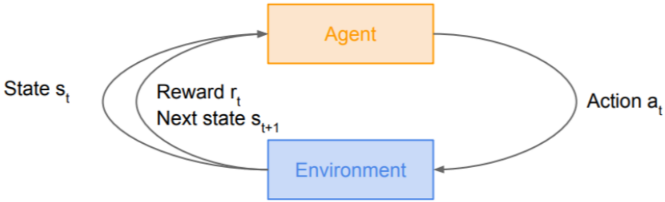
Outline

- Introduction to RL
- Proposal: Dual Agent RL (DARL)
  - Formalization and Objective
  - Connection to Dynamic Games
  - Algorithm Proposal: Policy Gradient
- Example Game: Lion and Zebra
- RL Simulation Results

1. Traditional RL

Optimizing Agent Behavior

Define interactions between an agent and its environment



Fundamentals of Reinforcement Learning

When action a is chosen from state s:
Transition to state s' with reward r with probability $p(s', r|s, a)$

From state s, agent chooses action a with probability $\pi(a|s)$

Reward from trajectory $\tau = (s_0, a_0, r_0, s_1, \dots)$ is $R_\tau = \sum_{k=0}^{|\tau|} r(s_k, a_k)$

Goal: Learn a policy π that maximizes $J(\pi) = \mathop{\mathbb{E}}_{\tau \sim \pi}(R_\tau)$

Fundamentals of Reinforcement Learning

A policy π is paramaterized by a matrix θ e.g.

$$\theta = \begin{array}{cc} \begin{array}{ccc} a_1 & a_2 & a_3 \end{array} & \\ \begin{bmatrix} 0.2 & 0.5 & 0.3 \\ 0 & 0 & 1 \\ 0.6 & 0.4 & 0 \\ \vdots & \vdots & \vdots \end{bmatrix} & \begin{array}{c} s_1 \\ s_2 \\ s_3 \\ \vdots \end{array} \end{array}$$

Optimal policies are computed iteratively as

$$\theta_{n+1} = \alpha \nabla_\theta J(\pi_{\theta_n}) + \theta_n$$

An expression for the gradient can be derived as

$$\nabla_\theta J(\pi_\theta) = \mathop{\mathbb{E}}_{\tau \sim \pi_\theta} \left[\sum_{k=0}^{|\tau|} \nabla_\theta \log \pi_\theta(a_k | s_k) \right]$$

Dynamic Game Structure

Multi-stage game (K stages)

- x_k node at which the game enters the kth stage
- u_k action of P_1 at the kth stage
- d_k action of P_2 at the kth stage

$$x_{k+1} = f_k(x_k, u_k, d_k)$$

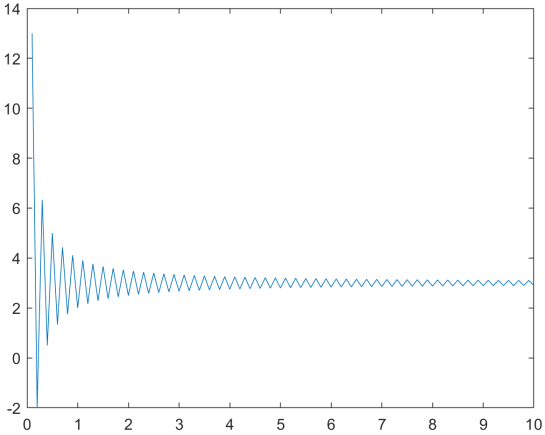
2. Dual-Agent Reinforcement Learning
(DARL)

DARL Background

- Two agents inhabit an environment
- Similar setup to zero-sum games
    - Two players: minimizer and maximizer
- Adversarial training
    - Increase robustness

Training by Bootstrapping

Inspiration: $J(\gamma^*, \sigma) \leq J(\gamma^*, \sigma^*) \leq J(\gamma, \sigma^*)$
Idea: Fix one agent and train the other and repeat until convergence



Parallels: DARL vs. Dynamic Games

DARL

- Stochastic
- Dual agents
- States $s_k \in S$
- Actions $a_k^1 \in A_1$, $a_k^2 \in A_2$
- Policies π^γ, π^σ
- Reward
  $R_\tau = \sum r(s_k, \pi^\gamma(s_k), \pi^\sigma(s_k))$
- P_1 wants to maximize
- P_2 wants to minimize

Dynamic Games

- Deterministic
- Two players
- States $x_k \in X$
- Actions $u_k \in \mathcal{U}$, $d_k \in \mathcal{D}$
- Policies γ, σ
- Outcome
  $J(\gamma, \sigma) = \sum g_k(x_k, \gamma(x_k), \sigma(x_k))$
- P_1 wants to minimize
- P_2 wants to maximize

Analogous concepts exist in both approaches

But reward R_τ is opposite of outcome $J(\gamma, \sigma)$

3. Example Game

Zebra and Lion

https://github.com/ezhang7423/game-theoretic-adversarial-rl/

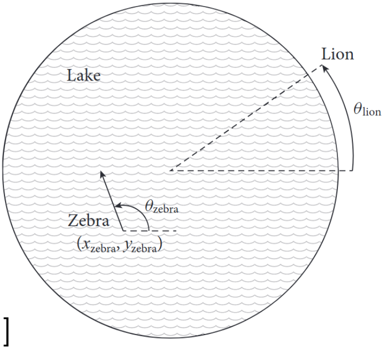
Overview

1. Define a game
2. Use game theory to find the optimal solution
3. Apply reinforcement learning and compare the simulated trajectory with the ideal agent behavior

Continuous Version

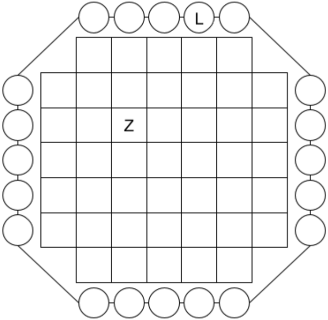
Zero-Sum Differential Game

- Zebra chooses $\theta_{zebra}(t) \in [0, 2\pi)$
- Lion chooses $\omega_{lion}(t) \in [-\frac{v_{lion}}{R}, +\frac{v_{lion}}{R}]$
- Outcome is $J = \begin{cases} T_{exit}, & \text{zebra escapes safely at time } T_{exit} \\ \infty, & \text{otherwise} \end{cases}$



Discrete Simplification

Zero-Sum Dynamic Game



- Zebra chooses $u_k \in \{N, E, S, W, X\}$
- Lion chooses $d_k \in \{-v_{max}, \ldots, +v_{max}\}$
- Outcome is $J = \begin{cases} K_{exit}, & \text{zebra escapes safely at stage } K_{exit} \\ \infty, & \text{otherwise} \end{cases}$
- States are given by $x_k = (x_{zebra}, y_{zebra}, pos_{lion})$

Analytic Solution

Game is solved by the recursive cost-to-go algorithm:

At each state x we compute

$$\begin{aligned} V_k(x) &= \min_u \max_d (1 + V_{k+1}(f(x,u,d))) \\ &= \max_d \min_u (1 + V_{k+1}(f(x,u,d))) \end{aligned}$$

with the boundary conditions

$$\begin{aligned} V_{K_{max}+1}(x) &= \infty, \quad \forall x \\ V_k(x) &= \infty, \quad \text{if zebra is caught in state } x \\ V_k(x) &= 0, \quad \text{if zebra escapes safely in state } x \end{aligned}$$

An example of optimal play:

```
 L
 --------      k=0
----------
-Z--------
----------
----------
----------
----------
----------
----------
 --------
```

An example of optimal play:

```
   L
   - - - - - - - -        k=1
 - - - - - - - - - -
 - - Z - - - - - - -
 - - - - - - - - - -
 - - - - - - - - - -
 - - - - - - - - - -
 - - - - - - - - - -
 - - - - - - - - - -
 - - - - - - - - - -
   - - - - - - - -
```

Analytic Solution

An example of optimal play:

```
   - - - - - - - -      k=2
  - - - - - - - - - -
  - - Z - - - - - - -
L - - - - - - - - - -
  - - - - - - - - - -
  - - - - - - - - - -
  - - - - - - - - - -
  - - - - - - - - - -
  - - - - - - - - - -
   - - - - - - - -
```

Analytic Solution

An example of optimal play:

```
 - - - - - - - -       k=3
- - - - - - - - - -
- - - Z - - - - - -
L- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
 - - - - - - - -
```

Analytic Solution

An example of optimal play:

```
   - - - - - - - -        k=4
  - - - - - - - - - -
  - - - Z - - - - - -
  - - - - - - - - - -
L - - - - - - - - - -
  - - - - - - - - - -
  - - - - - - - - - -
  - - - - - - - - - -
  - - - - - - - - - -
   - - - - - - - -
```

An example of optimal play:

```
  - - - - - - - - -      k=5
 - - - - - - - - - - -
 - - - - Z - - - - -
 - - - - - - - - - - -
L- - - - - - - - - - -
 - - - - - - - - - - -
 - - - - - - - - - - -
 - - - - - - - - - - -
 - - - - - - - - - - -
  - - - - - - - - -
```

Analytic Solution

An example of optimal play:

```
  - - - - - - - - -      k=6
 - - - - - - - - - - -
 - - - - Z - - - - -
L - - - - - - - - - - -
 - - - - - - - - - - -
 - - - - - - - - - - -
 - - - - - - - - - - -
 - - - - - - - - - - -
 - - - - - - - - - - -
  - - - - - - - - -
```

Analytic Solution

An example of optimal play:

```
  - - - - - - - - -      k=7
 - - - - - - - - - - -
 - - - - - - - - - - -
L- - - - Z- - - - -
 - - - - - - - - - - -
 - - - - - - - - - - -
 - - - - - - - - - - -
 - - - - - - - - - - -
 - - - - - - - - - - -
  - - - - - - - - -
```

Analytic Solution

An example of optimal play:

```
  --------      k=8
 ----------
 ----------
 ----Z-----
 ----------
L----------
 ----------
 ----------
 ----------
  --------
```

Analytic Solution

An example of optimal play:

```
  - - - - - - - -      k=9
- - - - - - - - - -
- - - - - - - - - -
- - - - -Z- - - -
- - - - - - - - - -
L- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
  - - - - - - - -
```

An example of optimal play:

```
  - - - - - - - -      k=10
 - - - - - - - - - -
L- - - - - - - - - -
 - - - - -Z- - - -
 - - - - - - - - - -
 - - - - - - - - - -
 - - - - - - - - - -
 - - - - - - - - - -
 - - - - - - - - - -
  - - - - - - - -
```

Analytic Solution

An example of optimal play:

```
  - - - - - - - -      k=11
 - - - - - - - - - -
L- - - - - - - - - -
 - - - - - - - - - -
 - - - - -Z- - - -
 - - - - - - - - - -
 - - - - - - - - - -
 - - - - - - - - - -
 - - - - - - - - - -
  - - - - - - - -
```

Analytic Solution

An example of optimal play:

```
 --------    k=12
L----------
 ----------
 ----------
 -----Z----
 ----------
 ----------
 ----------
 ----------
  --------
```

Analytic Solution

An example of optimal play:

```
     - - - - - - - - -        k=13
  L- - - - - - - - - - -
   - - - - - - - - - - -
   - - - - - - - - - - -
   - - - - - - - - - - -
   - - - - -Z- - - -
   - - - - - - - - - - -
   - - - - - - - - - - -
   - - - - - - - - - - -
     - - - - - - - - -
```

An example of optimal play:

```
   - - - - - - - -        k=14
  - - - - - - - - - -
  - - - - - - - - - -
  - - - - - - - - - -
 L- - - - - - - - - -
  - - - - -Z- - - -
  - - - - - - - - - -
  - - - - - - - - - -
  - - - - - - - - - -
   - - - - - - - -
```

An example of optimal play:

```
  - - - - - - - - -      k=15
 - - - - - - - - - - -
 - - - - - - - - - - -
 - - - - - - - - - - -
L- - - - - - - - - - -
 - - - - - - Z - - -
 - - - - - - - - - - -
 - - - - - - - - - - -
 - - - - - - - - - - -
  - - - - - - - - -
```

Analytic Solution

An example of optimal play:

```
   --------     k=16
  ----------
  ----------
  ----------
  ----------
  ------Z---
  ----------
 L----------
  ----------
   --------
```

Analytic Solution

An example of optimal play:

```
  - - - - - - - - -       k=17
 - - - - - - - - - - -
 - - - - - - - - - - -
 - - - - - - - - - - -
 - - - - - - - - - - -
 - - - - - - - Z - -
 - - - - - - - - - - -
L- - - - - - - - - - -
 - - - - - - - - - - -
  - - - - - - - - -
```

Analytic Solution

An example of optimal play:

```
  - - - - - - - - -        k=18
- - - - - - - - - - -
- - - - - - - - - - -
- - - - - - - - - - -
- - - - - - - - - - -
- - - - - - -Z- -
- - - - - - - - - - -
- - - - - - - - - - -
- - - - - - - - - - -
  - - - - - - - - -
   L
```

Analytic Solution

An example of optimal play:

```
  - - - - - - - - -      k=19
- - - - - - - - - - -
- - - - - - - - - - -
- - - - - - - - - - -
- - - - - - - - - - -
- - - - - - - - - Z -
- - - - - - - - - - -
- - - - - - - - - - -
- - - - - - - - - - -
  - - - - - - - - -
   L
```

Analytic Solution

An example of optimal play:

```
  - - - - - - - -       k=20
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - Z -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
  - - - - - - - -
         L
```

An example of optimal play:

```
 - - - - - - - - -      k=21
- - - - - - - - - - -
- - - - - - - - - - -
- - - - - - - - - - -
- - - - - - - - - - -
- - - - - - - - - -Z
- - - - - - - - - - -
- - - - - - - - - - -
- - - - - - - - - - -
 - - - - - - - - -
          L
```

Analytic Solution

An example of optimal play:

```
 - - - - - - - - -      k=22
- - - - - - - - - - -
- - - - - - - - - - -
- - - - - - - - - - -
- - - - - - - - - - -
- - - - - - - - - -Z
- - - - - - - - - - -
- - - - - - - - - - -
- - - - - - - - - - -
 - - - - - - - - -
                 L
```

Analytic Solution

An example of optimal play:

```
 - - - - - - - -     k=23
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -Z
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
 - - - - - - - -
                L
```

Analytic Solution

Stages for zebra to escape from each starting position:

```
[[nan 27.  1.  1.  1.  1.  1.  1.  1. nan]
 [ 1. 25. 23. 21. 19.  3.  3.  3.  3.  1.]
 [ 1. 23. 21. 19. 17. 17. 19.  5.  3.  1.]
 [ 1. 21. 19. 17. 15. 13.  7.  5.  3.  1.]
 [ 1.  3. 17. 15. 13. 11.  7.  5.  3.  1.]
 [ 1.  3. 17. 13. 11.  9.  7.  5.  3.  1.]
 [ 1.  3.  7.  7.  7.  7.  7.  5.  3.  1.]
 [ 1.  3.  5.  5.  5.  5.  5.  5.  3.  1.]
 [ 1.  3.  3.  3.  3.  3.  3.  3.  3.  1.]
 [nan  1.  1.  1.  1.  1.  1.  1.  1. nan]]
```

Analytic Solution: Simultaneous Play

We redefine the outcome as

$$J = \begin{cases} 0, & \text{zebra escapes safely before stage } K_{max} \\ 1, & \text{otherwise} \end{cases}$$

At each state x we compute

$$V_k(x) = \min_y \max_z y^T A z = \max_z \min_y y^T A z$$

$$\text{where} \qquad A_{ij} = V_{k+1}(f(x,i,j))$$

with the boundary conditions

$$\begin{aligned} V_{K_{max}+1}(x) &= 1, \quad \forall x \\ V_k(x) &= 1, \quad \text{if zebra is caught in state } x \\ V_k(x) &= 0, \quad \text{if zebra escapes safely in state } x \end{aligned}$$

Analytic Solution: Simultaneous Play

Probability that zebra is unable to escape within $K_{max} = 15$ stages:

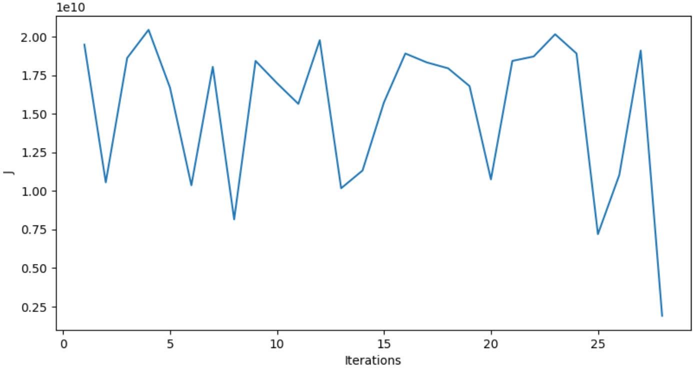
```
[[  nan 0.833 0.533 0.228 0.107 0.    0.    0.    0.      nan]
 [0.833 0.533 0.228 0.107 0.049 0.049 0.107 0.228 0.    0.   ]
 [0.533 0.228 0.107 0.049 0.022 0.014 0.049 0.107 0.    0.   ]
 [0.228 0.107 0.049 0.022 0.01  0.006 0.013 0.    0.    0.   ]
 [0.    0.049 0.022 0.01  0.004 0.    0.006 0.    0.    0.   ]
 [0.    0.049 0.014 0.006 0.    0.    0.    0.    0.    0.   ]
 [0.    0.107 0.049 0.012 0.    0.    0.    0.    0.    0.   ]
 [0.    0.    0.    0.    0.    0.    0.    0.    0.    0.   ]
 [0.    0.    0.    0.    0.    0.    0.    0.    0.    0.   ]
 [  nan 0.    0.    0.    0.    0.    0.    0.    0.      nan]]
```

4. RL Game Implementation

Zebra and Lion

RL Solution

Diverging J over training episodes



RL Solution

Using a gradient estimator: sampling all possible trajectories

$$\nabla_\theta \hat{J}(\pi_\theta) = \mathop{\mathbb{E}}_{\tau \sim D} \left[\sum_{k=0}^{|\tau|} \nabla_\theta \log \pi_\theta(a_k | s_k) \right]$$

RL Solution

An example of the learned policies:

```
 L
 - - -      k=0
- - - - -
- Z - - -
- - - - -
 - - -
```

RL Solution

An example of the learned policies:

```
  - - -      k=1
 - - - - -
L - - Z - -
 - - - - -
  - - -
```

RL Solution

An example of the learned policies:

```
  - - -     k=2
 - - Z - -
L - - - - -
 - - - - -
  - - -
```

RL Solution

An example of the learned policies:

```
  L
  - - -        k=3
- - Z - -
- - - - -
- - - - -
  - - -
```

RL Solution

An example of the learned policies:

```
  L
 - - -      k=4
- Z - - -
- - - - -
- - - - -
 - - -
```

RL Solution

An example of the learned policies:

```
 - - -      k=5
- - Z - - L
- - - - -
- - - - -
 - - -
```

RL Solution

An example of the learned policies:

```
  L
 - - -
- Z - - -
- - - - -
- - - - -
 - - -
```

k=6

RL Solution

An example of the learned policies:

```
  L
 - - -      k=7
Z- - - -
- - - - -
- - - - -
 - - -
```

Compare with the analytical results:

```
[[nan  1.   1.   1. nan]
 [ 1.  1.   1.   1.   1.]
 [ 1.  1.   1.   1.   0.]
 [ 1.  1.   1.   1.   0.]
 [nan  0.   0.   0. nan]]
```

Zebra had no chance of escape

Further Work

- Prove or disprove convergence of bootstrapping method
- Extend DARL to non zero-sum games
- Analyze properties of using the gradient estimator vs. true gradient

References

- Robust Adversarial Reinforcement Learning. http://proceedings.mlr.press/v70/pinto17a/pinto17a.pdf.
- An Overview of Multi-Agent Reinforcement ... - Arxiv.org. https://arxiv.org/pdf/2011.00583.pdf.
- “Game Theory and Multi-Agent Reinforcement Learning.” ResearchGate, https://www.researchgate.net/publication/269100101_Game_Theory_and_Multi-agent_Reinforcement_Learning.
- “CS231N: Convolutional Neural Networks for Visual Recognition.” Stanford University CS231n: Convolutional Neural Networks for Visual Recognition, http://cs231n.stanford.edu/.
- Openai. “Openai/Spinningup: An Educational Resource to Help Anyone Learn Deep Reinforcement Learning.” GitHub, https://github.com/openai/spinningup.