Review and Wrap Up

Professor Emma Brunskill

CS234 RL

Spring 2024

- Today the 3rd part of the lecture includes some slides from David Silver's introduction to RL slides or modifications of those slides

Where We Are In The Course & Reminders

- Last time: Quiz
- Today: Review and Looking Forward
- Wed 1-4pm Poster Session.
    - Location: AT&T Patio (Green space behind Computer Science Gates Building).
    - Reminder: Poster should also be uploaded before session. No late days.
- June 11 (Tuesday) 6pm: Final report due. No late days.

Today's Plan

- Quiz Recap
- Review and looking forward

-

- Quiz Recap
- Review and looking forward

Learning through experience/data to make good decisions under uncertainty

High Level Learning Goals[1]

- Define the key features of RL
- Given an application problem know how (and whether) to use RL for it
- Implement (in code) common RL algorithms
- Describe (list and define) multiple criteria for analyzing RL algorithms and evaluate algorithms on these metrics: e.g. regret, sample complexity, computational complexity, empirical performance, convergence, etc.
- Describe the exploration vs exploitation challenge and compare and contrast at least two approaches for addressing this challenge (in terms of performance, scalability, complexity of implementation, and theoretical guarantees)

---

[1]For more detailed descriptions, see website
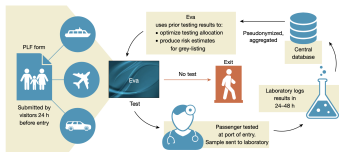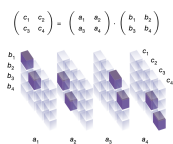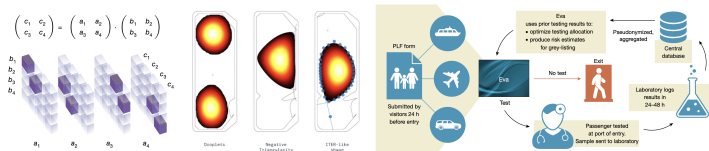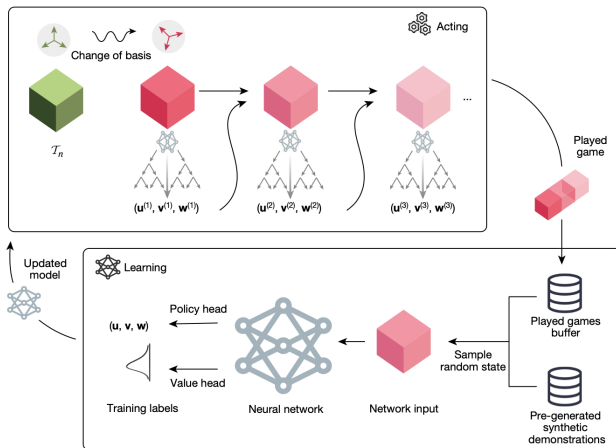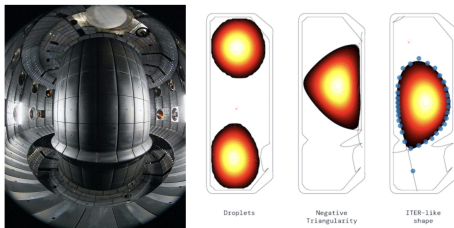
- Which domain are you choosing?
- Is this problem a bandit? A multi-step RL problem?
- Is the problem online / offline or some combination?
- What might the state / action / rewards be?
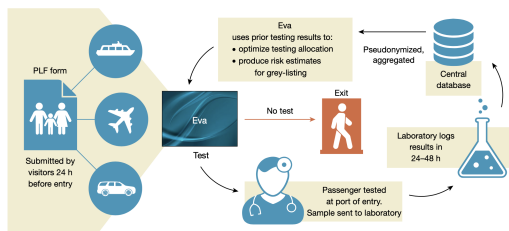- What algorithms might be useful here?

Droplets   Negative Triangularity   ITER-like shape

[3] Bastani et al. Nature 2021

Revisiting: ChatGPT (https://openai.com/blog/chatgpt/)

Step 1
Collect demonstration data and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3.5 with supervised learning.

Step 2
Collect comparison data and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Step 3
Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

A new prompt is sampled from the dataset.

The PPO model is initialized from the supervised policy.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

- Learn a policy $\pi(a|s)$ from data to optimize future expected reward
- Optimization, delayed consequences, exploration, generalization
- Actions impact data distribution: rewards observed and states reached

Reinforcement Learning: Standard Settings

- State dependence
  - Bandits: next state independent of prior state and action
  - General decision process: next state depends on prior states and actions
- Online/Offline
  - Offline / batch: Learn from historical data only
  - Online: Agent / algorithm can actively gather its own data

- Function approximation + Offpolicy learning is a key challenge
    - New policy introduces new distribution over (s,a,r)
    - Important because want data efficient RL in complex domains
    - PPO: Control with clipping
    - DAGGER: mitigate by obtaining more expert labels
    - Pessimistic Q Learning / CQL / MOPO: introduce pessimism into offline RL

---

[4]These align closely with many of the core points of Chelsea Finn's Deep RL course summary slides

- Function approximation + Offpolicy learning is a key challenge
  - New policy introduces new distribution over (s,a,r)
  - Important because want data efficient RL in complex domains
  - PPO: Control with clipping
  - DAGGER: mitigate by obtaining more expert labels
  - Pessimistic Q Learning / CQL / MOPO: introduce pessimism into offline RL
- Models, values and policies
  - Models: easier to represent uncertainty (why?), useful for MCTS
  - Q function: summarizes performance of policy & and implies policy
  - Policies: the main target of most RL applications
- Computational vs Data Efficiency
  - Data efficient techniques often very computationally intensive
  - In some domains, data = computation (e.g. simulated settings)

---

[5]These align closely with many of the core points of Chelsea Finn's Deep RL course summary slides

Open Challenges

- Practical, robust RL
  - Robust/stable: Need for automatic hyperparameter tuning, model selection, and generally robust methods for off-the-shelf RL
  - Efficiency: Need for data and computationally efficient methods
  - Hybrid offline-online:
- Framing the problem
  - Alternate formulations to Markov decision processes?
  - Multi-task vs single task?
  - Alternate forms of feedback?
  - Stochastic vs adversarial vs cooperative decision process?
  - Continuous learning + planning vs system identification then planning?
- Advancing data-driven decision making in domains that could benefit

Learning More

- CS224R Deep RL (Chelsea Finn)
- CS238 Decision Making under Uncertainty (Mykel Kochenderfer)
- CS239 / CS332 Advanced Decision Making Under Uncertainty / RL
- Ben Van Roy often offers an advanced class on bandits or RL

- Thanks for being part of the course!
- We look forward to your projects on Wednesday!