CS 598 Statistical Reinforcement Learning

Nan Jiang
What’s this course about?

• A grad-level seminar course on theory of RL
• with focus on sample complexity analyses
• all about proofs, some perspectives, 0 implementation
• Seminar course can be anywhere between students presenting papers and a rigorous course
  • In this course I will deliver most (or all) of the lectures
  • No text book; material is created by myself (course notes)
    • Related monograph under development w/ Sham Kakade and Alekh Agarwal
  • See course website for more material and references
Who should take this course?

• This course will be a good fit for you if you either
  • (A) have exposure to RL + comfortable with long mathematical derivations + interested in understanding RL from a purely theoretical perspective
  • (B) are proficient in learning theory and comfortable with highly abstract description of concepts / models / algorithms
• For people not in (A) or (B): I also teach CS498 RL (likely offered sp21), which is a lightweight version of this course
Prerequisites

• Maths
  • Linear algebra, probability & statistics, basic calculus
  • Markov chains
  • Optional: stochastic processes, numerical analysis
  • Useful: TCS background, empirical processes and statistical learning theory, optimization, online learning

• Exposure to ML
  • e.g., CS 446 Machine Learning
  • Experience with RL
Coursework

• Some readings after/before class
• Ad hoc homework to help digest particular material. Deadline will be lenient & TBA at the time of assignment
• Main assignment: course project (work on your own)
  • Baseline: reproduce theoretical analysis in existing papers
  • Advanced: identify an interesting/challenging extension to the paper and explore the novel research question yourself
  • Or, just work on a novel research question (must have a significant theoretical component; need to discuss with me)
Course project (cont.)

- See list of references and potential topics on website
- You will need to submit:
  - A brief proposal (~1/2 page). Tentative deadline: end of Oct
    - what’s the topic and what papers you plan to work on
    - why you choose the topic: what interest you?
    - which aspect(s) you will focus on?
  - Final report: clarity, precision, and brevity are greatly valued. More details to come…
- All docs should be in pdf. Final report should be prepared using LaTeX.
Course project (cont. 2)

Rule of thumb

1. learn something that interests you
2. teach me something! (I wouldn’t learn if I could not understand your report due to lack of clarity)
3. write a report similar to (or better than!) the notes I will share with you
Contents of the course

• many important topics in RL will not be covered in depth (e.g., TD). Read more (e.g., Sutton & Barto book) if you want to get a more comprehensive view of RL.

• the other opportunity to learn what’s not covered in lectures is the project, as potential topics for projects are much broader than what’s covered in class.
Logistics

• Course website: http://nanjiang.cs.illinois.edu/cs598/
  • logistics, links to slides/notes, and resources (e.g., textbooks to consult, related courses)
• Piazza for Q&A and announcements: see link on website.
  • Please self-enroll!
• Time: Wed & Fri 12:30-1:45pm.
• TA: Tengyang Xie
• Office hours: by appointment
Introduction to MDPs and RL
Reinforcement Learning (RL) Applications

[Levine et al’16] [Ng et al’03] [Singh et al’02] [Lei et al’12]
[Mandel et al’16] [Tesauro et al’07] [Mnih et al’15] [Silver et al’16]
Shortest Path

Bellman Equation

\[ V^*(d) = \min \{ 3 + V^*(g), 2 + V^*(f) \} \]

Greedy is suboptimal due to delayed effects

Need long-term planning
Shortest Path

Graph:
- Nodes: $s_0$, $b$, $c$, $d$, $e$, $f$, $g$
- Edges and Weights:
  - $s_0$ to $c$: 2
  - $b$ to $c$: 4
  - $s_0$ to $g$: 1
  - $b$ to $d$: 4
  - $d$ to $g$: 3
  - $c$ to $e$: 2
  - $d$ to $f$: 1
  - $g$ to $f$: 1
  - $e$ to $f$: 1

Path: $s_0$ -> $c$ -> $b$ -> $d$ -> $g$
Stochastic Shortest Path

Markov Decision Process (MDP)

State-Action Transition Distribution
Stochastic Shortest Path

Bellman Equation

\[ V^*(c) = \min \{ 4 + 0.7 \times V^*(d) + 0.3 \times V^*(e), 2 + V^*(e) \} \]

Greedy is suboptimal due to delayed effects

Need long-term planning
Stochastic Shortest Path via trial-and-error

Diagram:

- Start state: $s_0$
- States: $s_0, b, c, d, e, f, g$
- Edges with weights:
  - $s_0$ to $b$: 1
  - $s_0$ to $c$: 2
  - $b$ to $d$: 4
  - $c$ to $e$: 4
  - $d$ to $g$: 3
  - $d$ to $f$: 0.7
  - $f$ to $d$: 0.5
  - $f$ to $e$: 0.5
  - $g$ to $f$: 1
- Probabilities indicated by arrows:
  - $s_0$ to $b$: 0.3
  - $c$ to $e$: 0.3

The diagram illustrates a stochastic shortest path problem with trial-and-error as a method.
Stochastic Shortest Path via trial-and-error

Trajectory 1: $s_0 \searrow c \nearrow d \rightarrow g$

Trajectory 2:
Stochastic Shortest Path
via trial-and-error

Model-based RL

How many trajectories do we need to compute a near-optimal policy?

Sample complexity
Stochastic Shortest Path via trial-and-error

Nontrivial! Need exploration

- Assume states & actions are visited uniformly
- #trajectories needed \( \leq n \cdot (\text{#state-action pairs}) \)

How many trajectories do we need to compute a near-optimal policy?

#samples needed to estimate a multinomial distribution
Video game playing

reward \( r_t = R(s_t, a_t) \)

+20

e.g., random spawn of enemies

state \( s_t \in S \)

policy \( \pi: S \rightarrow A \)

action \( a_t \in A \)

transition dynamics \( P(\cdot|s_t, a_t) \)

(unknown)

objective: maximize \( \mathbb{E} \left[ \sum_{t=1}^{H} r_t | \pi \right] \)
Video game playing

Need generalization

Value function approximation
Video game playing

Need generalization

Value function approximation

Find $\theta$ s.t. $f(x;\theta) \approx r + \gamma \cdot E_{x'|x} [f(x';\theta)] \Rightarrow f(\cdot ; \theta) \approx V^*$
Adaptive medical treatment

- State: diagnosis
- Action: treatment
- Reward: progress in recovery
A Machine Learning view of RL
Lecture 1: Introduction to Reinforcement Learning

About RL

Many Faces of Reinforcement Learning

- Computer Science
- Economics
- Mathematics
- Engineering
- Neuroscience
- Psychology
- Machine Learning
- Classical/Operant Conditioning
- Bounded Rationality
- Reinforcement Learning
- Optimal Control
- Reward System
- Operations Research

slide credit: David Silver
Supervised Learning

Given \( \{(x^{(i)}, y^{(i)})\} \), learn \( f: x \mapsto y \)

- Online version: for round \( t = 1, 2, \ldots, \) the learner
  - observes \( x^{(t)} \)
  - predicts \( \hat{y}^{(t)} \)
  - receives \( y^{(t)} \)
- Want to maximize # of correct predictions
- e.g., classifies if an image is about a dog, a cat, a plane, etc. (multi-class classification)
- Dataset is fixed for everyone
- “Full information setting”
- Core challenge: generalization
Contextual bandits

For round $t = 1, 2, \ldots$, the learner

- Given $x_i$, chooses from a set of actions $a_i \in A$
- Receives reward $r(t) \sim R(x(t), a(t))$ (i.e., can be random)
- Want to maximize total reward
- You generate your own dataset $\{(x(t), a(t), r(t))\}$!
- e.g., for an image, the learner guesses a label, and is told whether correct or not (reward = 1 if correct and 0 otherwise). Do not know what’s the true label.
- e.g., for an user, the website recommends a movie, and observes whether the user likes it or not. Do not know what movies the user really want to see.
- “Partial information setting”
Contextual bandits

Contextual Bandits (cont.)

• Simplification: no $x$, Multi-Armed Bandits (MAB)
• Bandit is a research area by itself. we will not do a lot of bandits but may go through some material that have important implications on general RL (e.g., lower bounds)
For round $t = 1, 2, \ldots$,

- For time step $h=1, 2, \ldots, H$, the learner
  - Observes $x_h^{(t)}$
  - Chooses $a_h^{(t)}$
  - Receives $r_h^{(t)} \sim R(x_h^{(t)}, a_h^{(t)})$
  - Next $x_{h+1}^{(t)}$ is generated as a function of $x_h^{(t)}$ and $a_h^{(t)}$ (or sometimes, all previous $x$'s and $a$'s within round $t$)
- Bandits + “Delayed rewards/consequences”
- The protocol here is for episodic RL (each $t$ is an *episode*).
Why statistical RL?

Two types of scenarios in RL research

1. Solving a large planning problem using a learning approach
   - e.g., AlphaGo, video game playing, simulated robotics
   - Transition dynamics (Go rules) known, but too many states
   - Run the simulator to collect data

2. Solving a learning problem
   - e.g., adaptive medical treatment
   - Transition dynamics unknown (and too many states)
   - Interact with the environment to collect data
Why statistical RL?

Two types of scenarios in RL research

1. Solving a large planning problem using a learning approach

2. Solving a learning problem

- I am more interested in #2. More challenging in some aspects.
- Data (real-world interactions) is highest priority. Computation second.
- Even for #1, sample complexity lower-bounds computational complexity — statistical-first approach is reasonable.
  - caveat to this argument: you can do a lot more in a simulator; see http://hunch.net/?p=8825714
MDP Planning
Infinite-horizon discounted MDPs

An MDP $M = (S, A, P, R, \gamma)$

- State space $S$.
- Action space $A$.
- Transition function $P : S \times A \rightarrow \Delta(S)$. $\Delta(S)$ is the probability simplex over $S$, i.e., all non-negative vectors of length $|S|$ that sums up to 1
- Reward function $R : S \times A \rightarrow \mathbb{R}$. (deterministic reward function)
- Discount factor $\gamma \in [0,1)$

- The agent starts in some state $s_1$, takes action $a_1$, receives reward $r_2 \sim R(s_1, a_1)$, transitions to $s_2 \sim P(s_1, a_1)$, takes action $a_2$, so on so forth — the process continues indefinitely

We will only consider discrete and finite spaces in this course.
Value and policy

- Want to take actions in a way that maximizes value (or return):
  \[ \mathbb{E} \left[ \sum_{t=1}^{\infty} \gamma^{t-1} r_t \right] \]
  - This value depends on where you start and how you act
- Often assume boundedness of rewards: \( r_t \in [0, R_{\text{max}}] \)
  - What’s the range of \( \mathbb{E} \left[ \sum_{t=1}^{\infty} \gamma^{t-1} r_t \right] \) ? \( [0, \frac{R_{\text{max}}}{1 - \gamma}] \)
- A (deterministic) policy \( \pi: S \rightarrow A \) describes how the agent acts: at state \( s_t \), chooses action \( a_t = \pi(s_t) \).
- More generally, the agent may choose actions randomly (\( \pi: S \rightarrow \Delta(A) \)), or even in a way that varies across time steps (“non-stationary policies”)
- Define \( V^\pi(s) = \mathbb{E} \left[ \sum_{t=1}^{\infty} \gamma^{t-1} r_t \middle| s_1 = s, \pi \right] \)
Bellman equation for policy evaluation

\[
V^\pi(s) = \mathbb{E} \left[ \sum_{t=1}^{\infty} \gamma^{t-1} r_t \mid s_1 = s, \pi \right]
\]

\[
= \mathbb{E} \left[ r_1 + \sum_{t=2}^{\infty} \gamma^{t-1} r_t \mid s_1 = s, \pi \right]
\]

\[
= R(s, \pi(s)) + \sum_{s' \in S} P(s' \mid s, \pi(s)) \mathbb{E} \left[ \sum_{t=2}^{\infty} \gamma^{t-2} r_t \mid s_1 = s, s_2 = s', \pi \right]
\]

\[
= R(s, \pi(s)) + \sum_{s' \in S} P(s' \mid s, \pi(s)) \mathbb{E} \left[ \sum_{t=2}^{\infty} \gamma^{t-2} r_t \mid s_2 = s', \pi \right]
\]

\[
= R(s, \pi(s)) + \gamma \sum_{s' \in S} P(s' \mid s, \pi(s)) \mathbb{E} \left[ \sum_{t=1}^{\infty} \gamma^{t-1} r_t \mid s_1 = s', \pi \right]
\]

\[
= R(s, \pi(s)) + \gamma \sum_{s' \in S} P(s' \mid s, \pi(s)) V^\pi(s')
\]

\[
= R(s, \pi(s)) + \gamma \langle P(\cdot \mid s, \pi(s)), V^\pi(\cdot) \rangle
\]
Bellman equation for policy evaluation

\[ V^\pi(s) = R(s, \pi(s)) + \gamma \langle P(\cdot \mid s, \pi(s)), V^\pi(\cdot) \rangle \]

Matrix form: define

- \( V^\pi \) as the \(|S| \times 1\) vector \([V^\pi(s)]_{s \in S}\)
- \( R^\pi \) as the vector \([R(s, \pi(s))]_{s \in S}\)
- \( P^\pi \) as the matrix \([P(s' \mid s, \pi(s))]_{s \in S, s' \in S}\)

\[ V^\pi = R^\pi + \gamma P^\pi V^\pi \]

\[ (I - \gamma P^\pi)V^\pi = R^\pi \]

\[ V^\pi = (I - \gamma P^\pi)^{-1} R^\pi \]

This is always invertible. Proof?
State occupancy

\[(1 - \gamma) \cdot (I - \gamma P^\pi)^{-1}\]

Each row (indexed by \(s\)) is the normalized discounted state occupancy \(d^{\pi,s}\), whose \((s')\)-th entry is

\[d^{\pi,s}(s') = E \left[ \sum_{t=1}^{\infty} \gamma^{t-1} \mathbb{I}[s_t = s'] \left| s_1 = s, \pi \right. \right] \]

- \((1 - \gamma)\) is the normalization factor so that the matrix is row-stochastic.

- \(V^\pi(s)\) is the dot product between \(d^{\pi,s}/(1 - \gamma)\) and reward vector

- Can also be interpreted as the value function of indicator reward function
Optimality

• For infinite-horizon discounted MDPs, there always exists a stationary and deterministic policy that is optimal for all starting states simultaneously
  • Proof: Puterman’94, Thm 6.2.7 (reference due to Shipra Agrawal)

• Let $\pi^*$ denote this optimal policy, and $V^* := V^{\pi^*}$

• Bellman Optimality Equation:

\[
V^*(s) = \max_{a \in A} \left( R(s, a) + \gamma \mathbb{E}_{s' \sim P(s, a)} \left[ V^*(s') \right] \right)
\]

• If we know $V^*$, how to get $\pi^*$?

• Easier to work with Q-values: $Q^*(s, a)$, as $\pi^*(s) = \arg \max_{a \in A} Q^*(s, a)$

\[
Q^*(s, a) = R(s, a) + \gamma \mathbb{E}_{s' \sim P(s, a)} \left[ \max_{a' \in A} Q^*(s', a') \right]
\]
Ad Hoc Homework 1

- uploaded on course website
- help understand the relationships between alternative MDP formulations
- more like readings w/ questions to think about
- no need to submit