CS 498 Reinforcement Learning

Nan Jiang
Logistics

All information about this course: http://nanjiang.cs.illinois.edu/cs498/
Logistics

All information about this course: http://nanjiang.cs.illinois.edu/cs498/

- piazza link (please **self enroll**)
Logistics

All information about this course: http://nanjiang.cs.illinois.edu/cs498/

• piazza link (please **self enroll**!)
  • piazza will be the main platform for course-related communications
Logistics

All information about this course: http://nanjiang.cs.illinois.edu/cs498/

- piazza link (please **self enroll**)!
  - piazza will be the main platform for course-related communications
  - you can write private posts only visible to instructors, so post on piazza before requesting an individual meeting
Logistics

All information about this course: http://nanjiang.cs.illinois.edu/cs498/

• piazza link (please **self enroll**)!
  • piazza will be the main platform for course-related communications
  • you can write private posts only visible to instructors, so post on piazza before requesting an individual meeting
• Slides & schedule (updated every week)
Logistics

All information about this course: http://nanjiang.cs.illinois.edu/cs498/

• piazza link (please **self enroll**)!
  • piazza will be the main platform for course-related communications
  • you can write private posts only visible to instructors, so post on piazza before requesting an individual meeting

• Slides & schedule (updated every week)
• textbook links
Logistics

All information about this course: http://nanjiang.cs.illinois.edu/cs498/

- piazza link (please **self enroll**)!
  - piazza will be the main platform for course-related communications
  - you can write private posts only visible to instructors, so post on piazza before requesting an individual meeting
- Slides & schedule (updated every week)
- textbook links
- TA & office hours: 5-6pm Tue, 10-11am Fri (starting next week)
Logistics

All information about this course: http://nanjiang.cs.illinois.edu/cs498/

• piazza link (please **self enroll**)!
  • piazza will be the main platform for course-related communications
  • you can write private posts only visible to instructors, so post on piazza before requesting an individual meeting

• Slides & schedule (updated every week)
• textbook links
• TA & office hours: 5-6pm Tue, 10-11am Fri (starting next week)
• course policies
What’s this course about?
What’s this course about?

• A 400-level special topics course on RL
What’s this course about?

- A 400-level special topics course on RL
- Developed & taught for the first time — expect rough edges
What’s this course about?

• A 400-level special topics course on RL
• Developed & taught for the first time — expect rough edges
• Less theoretical version of CS598 that I taught last year
What’s this course about?

• A 400-level special topics course on RL
• Developed & taught for the first time — expect rough edges
• Less theoretical version of CS598 that I taught last year
• Will have some (minimum) programming components, but still lean heavily towards the mathematical side
What’s this course about?

- A 400-level special topics course on RL
- Developed & taught for the first time — expect rough edges
- Less theoretical version of CS598 that I taught last year
- Will have some (minimum) programming components, but still
  lean heavily towards the mathematical side
- Think of this mainly as a mathematical course; the programming
  component is just to ensure that you have some exposure to
  (some of) the actual algorithms
What’s this course about?

- A 400-level special topics course on RL
- Developed & taught for the first time — expect rough edges
- Less theoretical version of CS598 that I taught last year
- Will have some (minimum) programming components, but **still** lean heavily towards the mathematical side
- Think of this mainly as a mathematical course; the programming component is just to ensure that you have some exposure to (some of) the actual algorithms
- “I want to do cool applications” — we no **not** meet this demand (but hopefully will give you a good foundation for it)
What’s this course about?

- A 400-level special topics course on RL
- Developed & taught for the first time — expect rough edges
- Less theoretical version of CS598 that I taught last year
- Will have some (minimum) programming components, but still lean heavily towards the mathematical side
- Think of this mainly as a mathematical course; the programming component is just to ensure that you have some exposure to (some of) the actual algorithms
- “I want to do cool applications” — we no not meet this demand (but hopefully will give you a good foundation for it)
- Intellectual curiosity is the only reason you should take this course (and I will tell you why honestly in the class…)
Prerequisites

- Maths
Prerequisites

• Maths
  • Linear algebra, probability & statistics, basic calculus
Prerequisites

- **Maths**
  - Linear algebra, probability & statistics, basic calculus
  - Optional: Markov chains, stochastic processes, numerical analysis
Prerequisites

• Maths
• Linear algebra, probability & statistics, basic calculus
• Optional: Markov chains, stochastic processes, numerical analysis
• Exposure to ML
Prerequisites

• Maths
  • Linear algebra, probability & statistics, basic calculus
  • Optional: Markov chains, stochastic processes, numerical analysis

• Exposure to ML
  • e.g., CS 446 Machine Learning
Prerequisites

• Maths
  • Linear algebra, probability & statistics, basic calculus
  • Optional: Markov chains, stochastic processes, numerical analysis

• Exposure to ML
  • e.g., CS 446 Machine Learning

• We will soon have a homework 0 (hopefully this Thursday) that helps you recall some of the math tools and check whether you are mathematically prepared for this course
Coursework & Grading

- Some readings after/before class
Coursework & Grading

• Some readings after/before class
• Homework (50%): ~5 assignments, with both programming & writing components
Coursework & Grading

• Some readings after/before class
• Homework (50%): ~5 assignments, with both programming & writing components
  • Will likely to be handled via compass—more on this later
Coursework & Grading

- Some readings after/before class
- Homework (50%): ~5 assignments, with both programming & writing components
  - Will likely to be handled via compass—more on this later
- Exam (35%): Either final or a late mid exam
Coursework & Grading

• Some readings after/before class

• Homework (50%): ~5 assignments, with both programming & writing components
  • Will likely to be handled via compass—more on this later

• Exam (35%): Either final or a late mid exam

• Participation (15%): *might* take the form of in-class pop quizzes (without announcements)
Coursework & Grading

• Some readings after/before class
• Homework (50%): ~5 assignments, with both programming & writing components
  • Will likely to be handled via compass—more on this later
• Exam (35%): Either final or a late mid exam
• Participation (15%): *might* take the form of in-class pop quizzes (without announcements)
• For 4 credit students: an additional course project (reproducing part of an empirical or theory paper)
Introduction to MDPs and RL
Reinforcement Learning (RL) Applications
Reinforcement Learning (RL) Applications

[Levine et al’16]  [Ng et al’03]
Reinforcement Learning (RL) Applications

[Levine et al’16] [Ng et al’03] [Singh et al’02]
Reinforcement Learning (RL) Applications

[Levine et al’16] [Ng et al’03] [Singh et al’02] [Lei et al’12]
Reinforcement Learning (RL) Applications

[Levine et al’16] [Ng et al’03] [Singh et al’02] [Lei et al’12]
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]
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
Shortest Path

State

Action

\begin{itemize}
\item $s_0$
\item $b$
\item $c$
\item $d$
\item $e$
\item $f$
\item $g$
\end{itemize}

\begin{itemize}
\item 1
\item 2
\item 3
\item 4
\item 1
\item 3
\item 1
\end{itemize}
Greedy is suboptimal due to delayed effects

Need long-term planning
Greedy is suboptimal due to delayed effects

Need **long-term planning**
Greedy is suboptimal due to delayed effects

Need long-term planning

Bellman Equation \( V^*(d) = \min\{3 + V^*(g), 2 + V^*(f)\} \)
Greedy is suboptimal due to delayed effects

Need long-term planning

Bellman Equation
\[ V^*(d) = \min\{3 + V^*(g), 2 + V^*(f)\} \]
Greedy is suboptimal due to delayed effects

Need long-term planning
Greedy is suboptimal due to delayed effects

Need long-term planning
Shortest Path
Stochastic Shortest Path
Stochastic Shortest Path

Markov Decision Process (MDP)

State transition distribution

state

action

$S_0$

$B$

$C$

$D$

$E$

$F$

$G$

Transition probabilities:

- $s_0 \rightarrow b$: $0.3$
- $b \rightarrow d$: $0.7$
- $d \rightarrow f$: $0.5$
- $f \rightarrow g$: $0.5$
- $s_0 \rightarrow c$: $0.3$
- $c \rightarrow e$: $0.5$
- $e \rightarrow f$: $0.5$

Actions:

- $s_0 \rightarrow c$: Action 2
- $b \rightarrow d$: Action 4
- $d \rightarrow f$: Action 3
- $f \rightarrow g$: Action 1

States:

- $S_0$
- $B$
- $C$
- $D$
- $E$
- $F$
- $G$

Actions:

- 1
- 2
- 4
- 3
- 1
Stochastic Shortest Path

Greedy is suboptimal due to delayed effects

Need **long-term planning** (also known as *temporal credit assignment* in RL)
**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
(also known as temporal credit assignment in RL)
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
(also known as \textit{temporal credit assignment} in RL)
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
(also known as temporal credit assignment in RL)
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
(also known as temporal credit assignment in RL)
Stochastic Shortest Path
via trial-and-error
Stochastic Shortest Path via trial-and-error

Trajectory 1:

$s_0 \rightarrow c \rightarrow d \rightarrow g$

$s_0$
Stochastic Shortest Path
via trial-and-error

\( s_0 \rightarrow c \)
Stochastic Shortest Path via trial-and-error

s₀ ↘

s₀ ↘ c ↗

b

4

1

c

4

2

d

3

1

e

1

0.7

0.3

0.5

0.5

f

1

g

0.5

1

0.5

0.3

0.7
Stochastic Shortest Path via trial-and-error
Stochastic Shortest Path via trial-and-error

\[ s_0 \rightarrow c \rightarrow d \rightarrow g \]
Stochastic Shortest Path via trial-and-error

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

Trajectory 1: \(s_0 \searrow c \nearrow d \rightarrow g\)

Trajectory 2: \(s_0 \searrow c \nearrow e \rightarrow f \nearrow g\)
Stochastic Shortest Path
via trial-and-error

Trajectory 1: \( s_0 \downarrow c \uparrow d \rightarrow g \)

Trajectory 2: \( s_0 \downarrow c \uparrow e \rightarrow f \uparrow g \)

\[ \cdots \]
Stochastic Shortest Path via trial-and-error

Trajectory 1: $s_0 \downarrow c \uparrow d \rightarrow g$

Trajectory 2: $s_0 \downarrow c \uparrow e \rightarrow f \uparrow g$

...
Stochastic Shortest Path
via trial-and-error

Trajectory 1: \( s_0 \rightarrow c \rightarrow d \rightarrow g \)

Trajectory 2: \( s_0 \rightarrow c \rightarrow e \rightarrow f \rightarrow g \)

\[ \ldots \]
Stochastic Shortest Path
via trial-and-error

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

Trajectory 2: $s_0 \searrow c \nearrow e \rightarrow f \nearrow g$

...
Stochastic Shortest Path
via trial-and-error

Model-based RL

Trajectory 1: \( s_0 \searrow c \nearrow d \rightarrow g \)

Trajectory 2: \( s_0 \searrow c \nearrow e \rightarrow f \nearrow g \)

\[
\begin{array}{c}
0.72 \\
0.28 \\
0.55 \\
0.45
\end{array}
\]
Stochastic Shortest Path via trial-and-error

- Assume states & actions are visited uniformly
• Assume states & actions are visited uniformly
• As we have more data, the estimated distributions will be closer to true distributions, and the learned policy will be closer to optimal

Stochastic Shortest Path via trial-and-error
Stochastic Shortest Path
via trial-and-error

- Assume states & actions are visited uniformly
- As we have more data, the estimated distributions will be closer to true distributions, and the learned policy will be closer to optimal
- Sample complexity: the amount of data needed to guarantee that a certain level of near-optimality is achieved
Stochastic Shortest Path via trial-and-error

• Assume states & actions are visited uniformly
• As we have more data, the estimated distributions will be closer to true distributions, and the learned policy will be closer to optimal
• Sample complexity: the amount of data needed to guarantee that a certain level of near-optimality is achieved
- Assume states & actions are visited uniformly
- As we have more data, the estimated distributions will be closer to true distributions, and the learned policy will be closer to optimal
- Sample complexity: the amount of data needed to guarantee that a certain level of near-optimality is achieved
• Assume states & actions are visited uniformly
• As we have more data, the estimated distributions will be closer to true distributions, and the learned policy will be closer to optimal
• Sample complexity: the amount of data needed to guarantee that a certain level of near-optimality is achieved
Nontrivial! Need exploration

- Assume states & actions are visited uniformly
- As we have more data, the estimated distributions will be closer to true distributions, and the learned policy will be closer to optimal
- Sample complexity: the amount of data needed to guarantee that a certain level of near-optimality is achieved
Stochastic Shortest Path
via trial-and-error

Nontrivial! Need exploration

- Assume states & actions are visited uniformly
- As we have more data, the estimated distributions will be closer to true distributions, and the learned policy will be closer to optimal
- Sample complexity: the amount of data needed to guarantee that a certain level of near-optimality is achieved
Random exploration can be inefficient
Random exploration can be inefficient
Random exploration can be inefficient
Random exploration can be inefficient visited in $2^{-H}$ fraction of all trajectories
Random exploration can be inefficient

visited in $2^{-H}$ fraction of all trajectories

Freeway (one of the Atari games)

“Freeway + RL”: https://youtu.be/44CilPmlimQ
Random exploration can be inefficient

visited in $2^{-H}$ fraction of all trajectories

Freeway (one of the Atari games)

“Freeway + RL”: https://youtu.be/44CilPmlimQ
Video game playing

state $s_t \in S$
Video game playing

state $s_t \in S$

action $a_t \in A$

reward $r_t = R(s_t, a_t)$

+20
Video game playing

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

state \( s_t \in S \)

action \( a_t \in A \)

e.g., random spawn of enemies

transition dynamics \( P(\cdot | s_t, a_t) \) (unknown)
Video game playing

- State: $s_t \in S$
- Action: $a_t \in A$
- Reward: $r_t = R(s_t, a_t)$
- 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

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

e.g., random spawn of enemies

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

Need generalization
Video game playing

Need generalization

Value function approximation
Video game playing

Neural network architecture

$f(x;\theta)$

state features $x$

Need generalization

Value function approximation
Video game playing

Need generalization
Value function approximation
Video game playing

Find $\theta$ s.t.

$\mathbf{f}(x; \theta) \approx r + \gamma \cdot E_{x'}|x\left[ f(x'; \theta) \right]$
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
Adaptive medical treatment

- State: diagnosis
- Action: treatment
- Reward: progress in recovery

Formulating a real problem in the RL framework is difficult, and defining your state, action, horizon, and reward can be tricky—will come back later in the course.
Summary of this Part: 3 core challenges of RL
Summary of this Part: 3 core challenges of RL

• Temporal credit assignment
Summary of this Part: 3 core challenges of RL

- Temporal credit assignment
  - A sequence of actions led to success/failure: which action(s) to attribute the consequence to?
Summary of this Part: 3 core challenges of RL

• Temporal credit assignment
  • A sequence of actions led to success/failure: which action(s) to attribute the consequence to?
• Exploration
Summary of this Part: 3 core challenges of RL

- Temporal credit assignment
  - A sequence of actions led to success/failure: which action(s) to attribute the consequence to?
- Exploration
  - How to take actions during data collection to guarantee that the dataset provide a comprehensive description of the environment?
Summary of this Part: 3 core challenges of RL

• Temporal credit assignment
  • A sequence of actions led to success/failure: which action(s) to attribute the consequence to?

• Exploration
  • How to take actions during data collection to guarantee that the dataset provide a comprehensive description of the environment?

• Generalization
Summary of this Part: 3 core challenges of RL

- Temporal credit assignment
  - A sequence of actions led to success/failure: which action(s) to attribute the consequence to?
- Exploration
  - How to take actions during data collection to guarantee that the dataset provide a comprehensive description of the environment?
- Generalization
  - How do deal with (very) large state spaces?
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

Machine Learning

Optimal Control

Reinforcement Learning

Reward System

Operations Research

Classical/Operant Conditioning

Bounded Rationality

Psychology

Economics

slide credit: David Silver
Supervised Learning

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

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

- Online version: for round \( t = 1, 2, \ldots \), the learner
Supervised Learning

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

- Online version: for round \( t = 1, 2, \ldots \), the learner
  - observes \( x^{(t)} \)
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 \( \hat{y}^{(t)} \)
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 \(y^{(t)}\)
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
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)
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
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”
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
Contextual bandits

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

- Given $x^{(t)}$, chooses from a set of actions $a^{(t)} \in A$
Contextual bandits

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

• Given $x(t)$, chooses from a set of actions $a(t) \in A$
• Receives reward $r(t) \sim R(x(t), a(t))$ (i.e., can be random)
Contextual bandits

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

1. Given $x^{(t)}$, chooses from a set of actions $a^{(t)} \in A$
2. Receives reward $r^{(t)} \sim R(x^{(t)}, a^{(t)})$ (i.e., can be random)
3. Want to maximize total reward
Contextual bandits

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

- Given $x(t)$, chooses from a set of actions $a(t) \in A$
- 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 $\{(x^{(t)}, a^{(t)}, r^{(t)})\}$!
Contextual bandits

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

- Given $x^{(t)}$, chooses from a set of actions $a^{(t)} \in A$
- 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.
Contextual bandits

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

- Given $x^{(t)}$, chooses from a set of actions $a^{(t)} \in A$
- 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.**
Contextual bandits

For round \( t = 1, 2, \ldots \), the learner

- Given \( x(t) \), chooses from a set of actions \( a(t) \in A \)
- 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.)
Contextual bandits

Contextual Bandits (cont.)

- Simplification: no $x$, Multi-Armed Bandits (MAB)
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 $t = 1, 2, \ldots$, 

RL
For round $t = 1, 2, \ldots$,

- For time step $h=1, 2, \ldots, H$, the learner
For round $t = 1, 2, \ldots,$

- For time step $h=1, 2, \ldots, H$, the learner
  - Observes $x_h^{(t)}$
For round $t = 1, 2, \ldots$,

- For time step $h=1, 2, \ldots, H$, the learner
  - Observes $x_h(t)$
  - Chooses $a_h(t)$
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 $r_{h(t)} \sim R(x_{h(t)}, a_{h(t)})$
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 $t$)
For round $t = 1, 2, ...$

- For time step $h=1, 2, ..., H$, 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”
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).