CS394R
Reinforcement Learning:
Theory and Practice

Scott Niekum and Peter Stone

Department of Computer Science
The University of Texas at Austin
Good Morning Colleagues
Good Morning Colleagues

- Are there any questions?
Logistics

- Resources page - and Sutton materials
Logistics

- Resources page - and Sutton materials
- Next week’s readings
Chapter 3

- Defined the problem
Chapter 3

- Defined the problem
- Introduced some important notation and concepts.
Chapter 3

• Defined the problem

• Introduced some important notation and concepts.
  – Returns
  – Markov property
  – State/action value functions
  – Bellman equations
Chapter 3

• Defined the problem

• Introduced some important notation and concepts.
  – Returns
  – Markov property
  – State/action value functions
  – Bellman equations
  – Get comfortable with them!
Chapter 3

• Defined the problem

• Introduced some important notation and concepts.
  – Returns
  – Markov property
  – State/action value functions
  – Bellman equations
  – Get comfortable with them!
    – \( q_\pi(s, a) = \ldots \)
Chapter 3

- Defined the problem

- Introduced some important notation and concepts.
  - Returns
  - Markov property
  - State/action value functions
  - Bellman equations
  - Get comfortable with them!
  - $q_\pi(s, a) = \ldots$ (Exercise 3.13)
Chapter 3

- Defined the problem
- Introduced some important notation and concepts.
  - Returns
  - Markov property
  - State/action value functions
  - Bellman equations
  - Get comfortable with them!
    - $q_\pi(s, a) = \ldots$ (Exercise 3.13)
    - Backup diagrams
Chapter 3

- Defined the problem

- Introduced some important notation and concepts.
  - Returns
  - Markov property
  - State/action value functions
  - Bellman equations
  - Get comfortable with them!
    - $q_\pi(s, a) = \ldots$ (Exercise 3.13)
    - Backup diagrams

- Solution methods start in Chapter 4
Chapter 3

- Defined the problem
- Introduced some important notation and concepts.
  - Returns
  - Markov property
  - State/action value functions
  - Bellman equations
  - Get comfortable with them!
    - $q_\pi(s, a) = \ldots$ (Exercise 3.13)
    - Backup diagrams
- Solution methods start in Chapter 4
  - What does it mean to solve an RL problem?
Formulating the RL problem

- Art more than science
Formulating the RL problem

- Art more than science
- States, actions, rewards
Formulating the RL problem

- Art more than science
- States, actions, rewards
  - Rewards: no hints on how to solve the problem
Formulating the RL problem

- Art more than science
- States, actions, rewards
  - Rewards: no hints on how to solve the problem
  - Joseph Muffoletto: in chess, doesn’t that make the problem very hard to solve?
Formulating the RL problem

- Art more than science
- States, actions, rewards
  - Rewards: no hints on how to solve the problem
  - Joseph Muffoletto: in chess, doesn’t that make the problem very hard to solve?
- Discount factor part of the environment
Markov property

• Does it hold in the real world?
Markov property

- Does it hold in the real world?
  - Nikita Gollamudi: Are any systems "fundamentally" non-Markovian?
Markov property

- Does it hold in the real world?
  - Nikita Gollamudi: Are any systems "fundamentally" non-Markovian?
  - What if there’s a time horizon?
Markov property

- Does it hold in the real world?
  - Nikita Gollamudi: Are any systems "fundamentally" non-Markovian?
  - What if there’s a time horizon?

- It’s an ideal
  - Will allow us to prove properties of algorithms
Markov property

- Does it hold in the real world?
  - Nikita Gollamudi: Are any systems "fundamentally" non-Markovian?
  - What if there’s a time horizon?

- It’s an ideal
  - Will allow us to prove properties of algorithms
  - Algorithms may still work when not provably correct
Markov property

- Does it hold in the real world?
  - Nikita Gollamudi: Are any systems "fundamentally" non-Markovian?
  - What if there’s a time horizon?

- It’s an ideal
  - Will allow us to prove properties of algorithms
  - Algorithms may still work when not provably correct
  - Could you compensate? Do algorithms change?
Markov property

- Does it hold in the real world?
  - Nikita Gollamudi: Are any systems "fundamentally" non-Markovian?
  - What if there’s a time horizon?

- It’s an ideal
  - Will allow us to prove properties of algorithms
  - Algorithms may still work when not provably correct
  - Could you compensate? Do algorithms change?
  - If not, you may want different algorithms (Monte Carlo)
Chapter 4

- Solution methods *given a model*
• Solution methods *given a model*
  – So no exploration vs. exploitation
• Solution methods *given a model*
  – So no exploration vs. exploitation

• Use *bootstrapping*
Policy Evaluation

- $V^\pi$ exists and is unique if $\gamma < 1$ or termination guaranteed for all states under policy $\pi$. 
Policy Evaluation

- $V^\pi$ exists and is unique if $\gamma < 1$ or termination guaranteed for all states under policy $\pi$.
- Policy evaluation converges under the same conditions.
Policy Evaluation

- $V^\pi$ exists and is unique if $\gamma < 1$ or termination guaranteed for all states under policy $\pi$.
- Policy evaluation converges under the same conditions.
- Policy evaluation on the week 1 problem
  - undiscounted, episodic
Policy Evaluation

- $V^\pi$ exists and is unique if $\gamma < 1$ or termination guaranteed for all states under policy $\pi$.

- Policy evaluation converges under the same conditions.

- Policy evaluation on the week 1 problem
  - undiscounted, episodic
  - Are the conditions met?
• Policy improvement theorem:

\[ \forall s, q_\pi(s, \pi'(s)) \geq v_\pi(s) \Rightarrow \forall s, v_{\pi'}(s) \geq v_\pi(s) \]
Policy Improvement

- **Policy improvement theorem:**
  \[ \forall s, q_{\pi}(s, \pi'(s)) \geq v_{\pi}(s) \Rightarrow \forall s, v_{\pi'}(s) \geq v_{\pi}(s) \]

- **Polynomial time convergence** (in number of states \( n \) and actions \( m \)) even though \( m^n \) policies.
  - Ignoring effect of \( \gamma \) and bits to represent rewards/transitions.
Value Iteration on Week 1 problem

- Show the new policy at each step
  - Doesn’t actually compute policy
Value Iteration on Week 1 problem

- Show the new policy at each step
  - Doesn’t actually compute policy
  - Break policy ties with equiprobable actions
Value Iteration on Week 1 problem

- Show the new policy at each step
  - Doesn’t actually compute policy
  - Break policy ties with equiprobable actions
  - No stochastic transitions
Value Iteration on Week 1 problem

- Show the new policy at each step
  - Doesn’t actually compute policy
  - Break policy ties with equiprobable actions
  - No stochastic transitions

- How would policy iteration proceed in comparison?
  - More or fewer policy updates?
Value Iteration on Week 1 problem

- Show the new policy at each step
  - Doesn’t actually compute policy
  - Break policy ties with equiprobable actions
  - No stochastic transitions

- How would policy iteration proceed in comparison?
  - More or fewer policy updates?
  - True in general?
Value Iteration on Week 1 problem

• Show the new policy at each step
  – Doesn’t actually compute policy
  – Break policy ties with equiprobable actions
  – No stochastic transitions

• How would policy iteration proceed in comparison?
  – More or fewer policy updates?
  – True in general?

• How important are the initial values?
Interesting Questions

- Oguzhan Akcin: Is it possible for DP to get stuck in the local minimum?
Interesting Questions

- Oguzhan Akcin: Is it possible for DP to get stuck in the local minimum?

- Jiaxun Cui: If we are not able to visit each state at least once, do PE and PI find an optimal policy?
Interesting Questions

- Oguzhan Akcin: Is it possible for DP to get stuck in the local minimum?

- Jiaxun Cui: If we are not able to visit each state at least once, do PE and PI find an optimal policy?

- Caroline Wang: Why treat prediction and control separately? Why is the prediction problem important?
Interesting Questions

- Oguzhan Akcin: Is it possible for DP to get stuck in the local minimum?
- Jiaxun Cui: If we are not able to visit each state at least once, do PE and PI find an optimal policy?
- Caroline Wang: Why treat prediction and control separately? Why is the prediction problem important?
- Stephane Hatgiskessell: When can asynchronous DP ignore states?
- Jeongmu Daniel Hahn: How can asynchronous DP reduce memory usage?
Chapter 4 Summary

- Chapter 4 treats **bootstrapping** with a **model**
Chapter 4 Summary

- Chapter 4 treats **bootstrapping with a model**
  - Next: no model and no bootstrapping
Chapter 4 Summary

- Chapter 4 treats **bootstrapping** with a **model**
  - Next: no model and no bootstrapping
  - Then: no model, but bootstrapping