Good Morning Colleagues

- Are there any questions?
Logistics

- Project feedback: mostly good, but consider revising
Logistics

- Project feedback: mostly good, but consider revising
- Please do the class midterm Survey - due Friday
Logistics

• Project feedback: mostly good, but consider revising
• Please do the class midterm Survey - due Friday
• More readings - coming soon
Options

• Extension of RL to temporal abstraction
Options

- Extension of RL to temporal abstraction
- State abstraction vs. temporal abstraction...
Options

• Extension of RL to temporal abstraction

• State abstraction vs. temporal abstraction...
  – ... Week 1 task!
Options

• Extension of RL to temporal abstraction

• State abstraction vs. temporal abstraction...
  - ... Week 1 task!
  - p. 14?
Options

- Extension of RL to temporal abstraction
- State abstraction vs. temporal abstraction...  
  - ... Week 1 task!
  - p. 14?
- They don’t address what temporal abstraction to use — they just show how it can fit into the RL formalism
Options

- Extension of RL to temporal abstraction
- State abstraction vs. temporal abstraction...
  - ... Week 1 task!
  - p. 14?
- They don’t address what temporal abstraction to use — they just show how it can fit into the RL formalism
  - Why couldn’t it before?
Options

• Extension of RL to temporal abstraction

• State abstraction vs. temporal abstraction...
  – ... Week 1 task!
  – p. 14?

• They don’t address what temporal abstraction to use — they just show how it can fit into the RL formalism
  – Why couldn’t it before?

• Markov vs. Semi-markov:
  – states, actions
  – mapping from (s, a) to expected discounted reward
  – well-defined distribution of next state, transit time
Discussion Points

- Are composed options *always* semi-Markov?
Discussion Points

- Are composed options *always* semi-Markov?
- What happens when initial value functions are optimistic? (slides)
Discussion Points

- Are composed options always semi-Markov?
- What happens when initial value functions are optimistic? (slides)
- Option discovery (slides)
Discussion Points

- Are composed options always semi-Markov?

- What happens when initial value functions are optimistic? (slides)

- Option discovery (slides)
  - bottleneck states
  - novelty
  - changed useful state abstractions (slides)
Discussion Points

- Are composed options always semi-Markov?

- What happens when initial value functions are optimistic? (slides)

- Option discovery (slides)
  - bottleneck states
  - novelty
  - changed useful state abstractions (slides)
MAXQ

- Defines how to learn given a task hierarchically
MAXQ

• Defines how to learn given a task hierarchically

• Does not address how to construct the hierarchy
MAXQ

- Defines how to learn given a task hierarchically
- Does not address how to construct the hierarchy
- Strives for recursive optimality
MAXQ

- Defines how to learn given a task hierarchically
- Does not address how to construct the hierarchy
- Strives for **recursive optimality**— local optimality given subtask policies
MAXQ

- Defines how to learn given a task hierarchically
- Does not address how to construct the hierarchy
- Strives for recursive optimality—local optimality given subtask policies
  - Class discussion
MAXQ

- Defines how to learn given a task hierarchically
- Does not address how to construct the hierarchy
- Strives for recursive optimality—local optimality given subtask policies
  - Class discussion
  - Weaker or stronger than hierarchical optimality?
MAXQ

- Defines how to learn given a task hierarchically
- Does not address how to construct the hierarchy
- Strives for *recursive optimality*— local optimality given subtask policies
  - Class discussion
  - Weaker or stronger than hierarchical optimality?
- Enables reuse of subtasks
MAXQ

- Defines how to learn given a task hierarchically
- Does not address how to construct the hierarchy
- Strives for **recursive optimality**— local optimality given subtask policies
  - Class discussion
  - Weaker or stronger than hierarchical optimality?
- Enables reuse of subtasks
- Enables useful state abstraction (how?)
Some details

- a means both primitive actions and subtasks (options)
Some details

- $a$ means both primitive actions and subtasks (options)

- Context-dependent vs. context-independent
Some details

- $a$ means both primitive actions and subtasks (options)
- Context-dependent vs. context-independent
- Higher-level subtasks are essentially policies over options
  - But subtasks are learned too
  - And the values propagate correctly
Some details

- \( a \) means both primitive actions and subtasks (options)
- Context-dependent vs. context-independent
- Higher-level subtasks are essentially policies over options
  - But subtasks are learned too
  - And the values propagate correctly
- What does \( C^\pi_i(s, a) \) mean?
Some details

• $a$ means both primitive actions and subtasks (options)

• Context-dependent vs. context-independent

• Higher-level subtasks are essentially policies over options
  – But subtasks are learned too
  – And the values propagate correctly

• What does $C^\pi_i(s,a)$ mean? (Nick slides)
Some details

- \( a \) means both primitive actions and subtasks (options)
- Context-dependent vs. context-independent
- Higher-level subtasks are essentially policies over options
  - But subtasks are learned too
  - And the values propagate correctly
- What does \( C_i^\pi(s, a) \) mean? (Nick slides)
- How does equation (2) relate to flat Q?
Some details

- \( a \) means both primitive actions and subtasks (options)

- Context-dependent vs. context-independent

- Higher-level subtasks are essentially policies over options
  - But subtasks are learned too
  - And the values propagate correctly

- What does \( C_i^{\pi}(s, a) \) mean? (Nick slides)

- How does equation (2) relate to flat Q?

- Polling: Why the dip in the graph in Figure 6?
Discussion Points

- What does MAXQ-Q buy you over flat?
Discussion Points

• What does MAXQ-Q buy you over flat?

• What does polling buy you over flat?