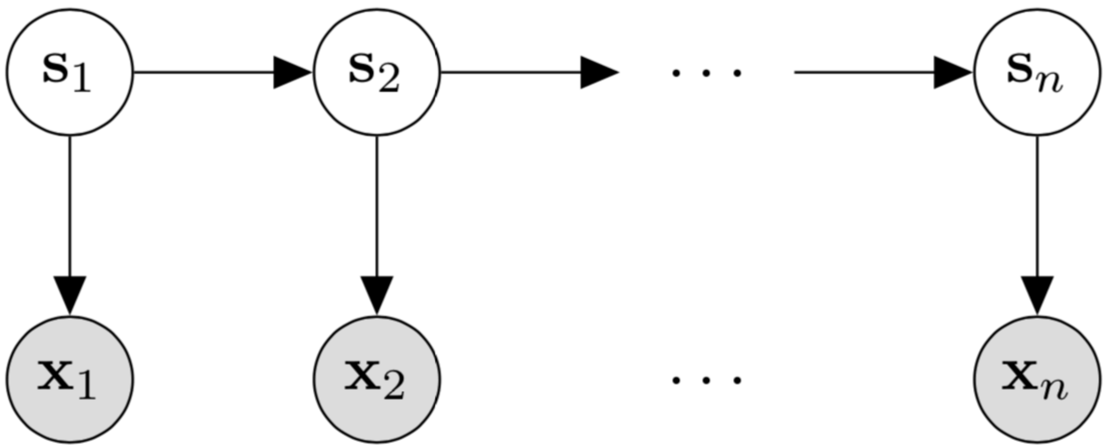
CS181: Introduction to Machine Learning

Lecture 20 (MDPs)

Spring 2021

Finale Doshi-Velez and David C. Parkes
Harvard Computer Science

Recap: Hidden Markov Models

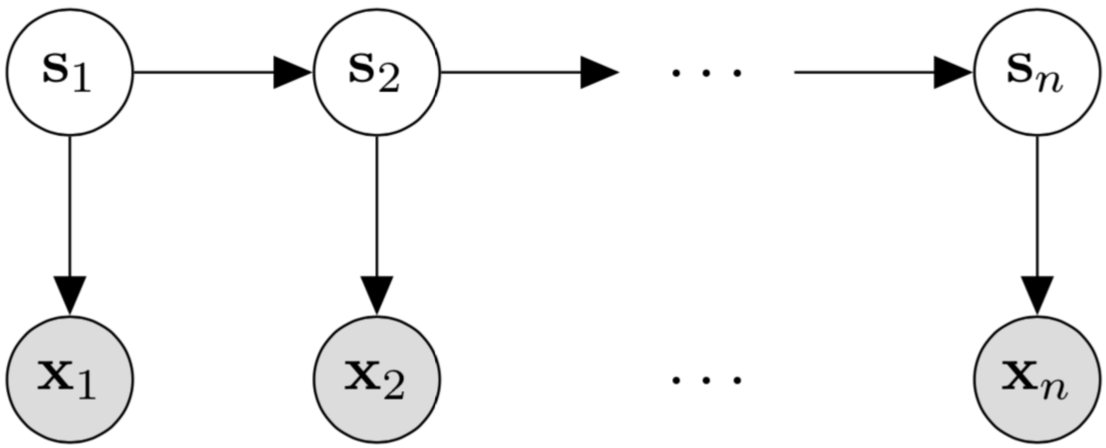


Application: Automated vehicles

- **State**: empty, parked, waiting, turning
- **Observation** (discrete, continuous): position, velocity, size, color, #passengers

Of interest: what is the probability another vehicle is parked given the sequence of observations?

Recap: Hidden Markov Models

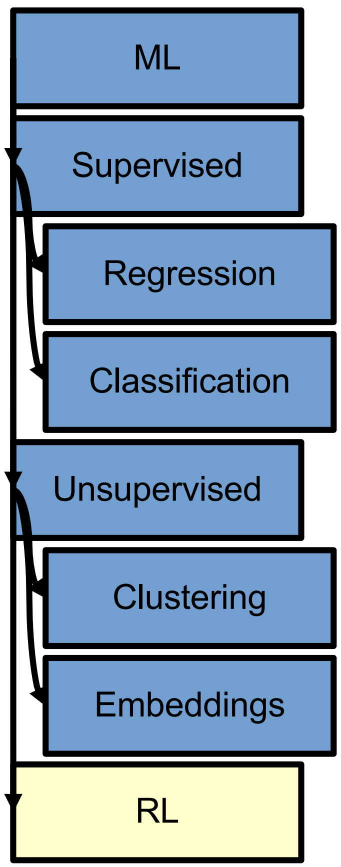


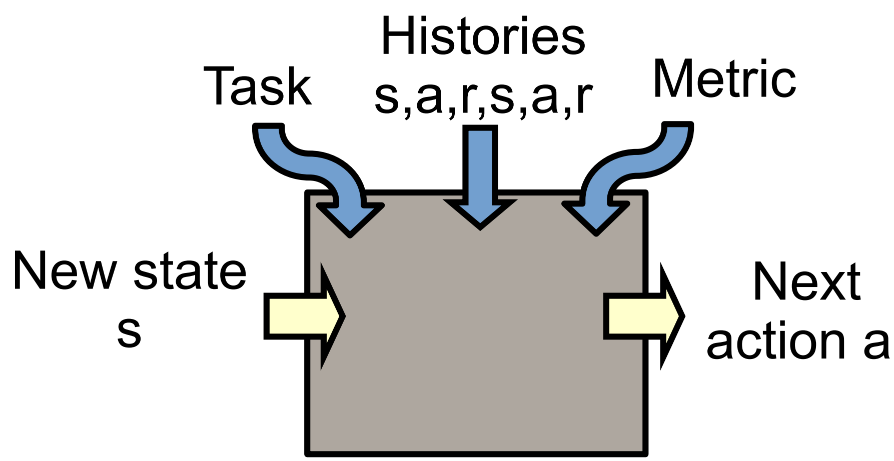
Application: Speech recognition

- **State**: (discrete): phoneme /e/ (elf), /m/ (mum), /n/ (name), /k/ (cat)
- **Observation** (continuous): frequency (e.g., a 10-dim, real vector); modeled via a mixture-of-Gaussians

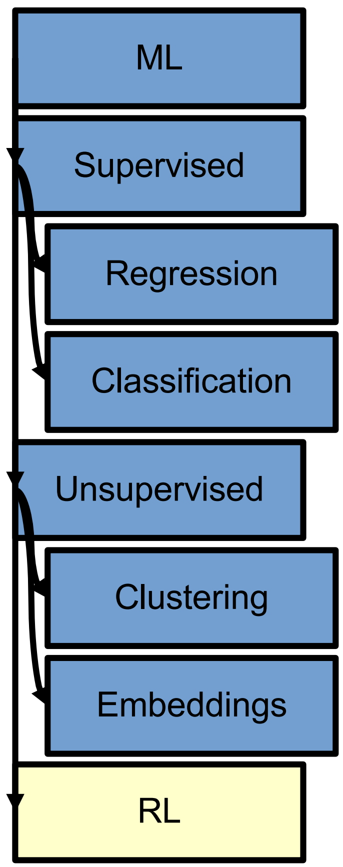
<u>Of interest</u>: what is the most likely sequence of phonemes, given the observations?

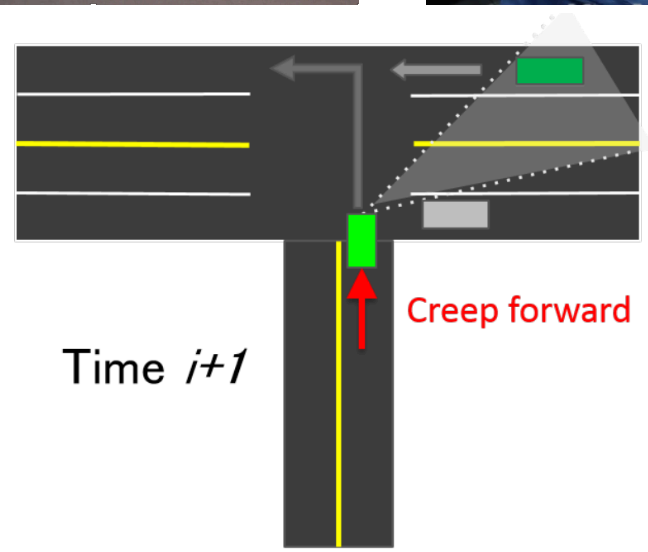
Terminology: Reinforcement Learning





Terminology: Reinforcement Learning





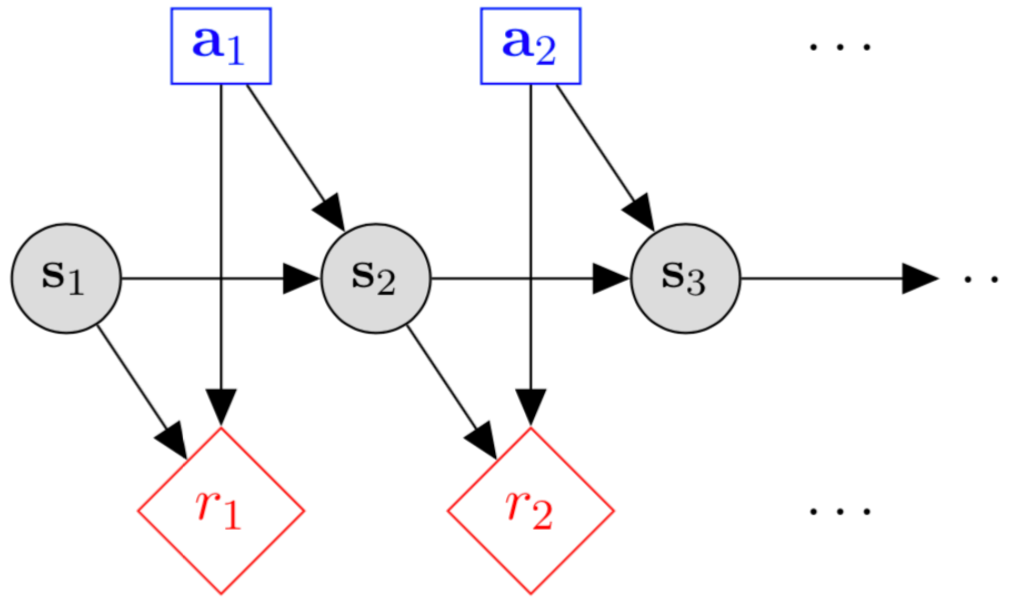
Kormushev et al., 2010 https://www.youtube.com/watch?v=W_gxLKSsSIE
Boston Dynamics. Isele et al. https://arxiv.org/pdf/1705.01196.pdf

Today: Learning to Act

<u>Learning to act:</u>

embodied agents

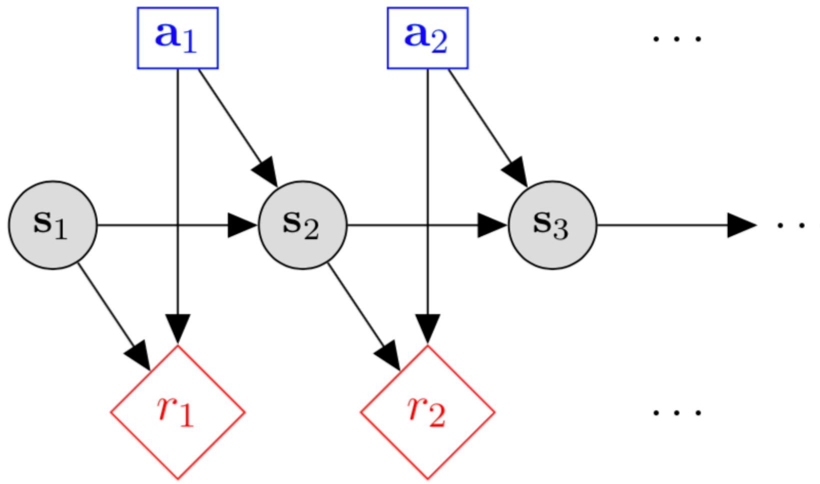
$$D = (s_1, a_1, r_1, s_2, a_2, r_2, \ldots)$$



state, action, reward, state, action, reward…
"**Markov decision process**"

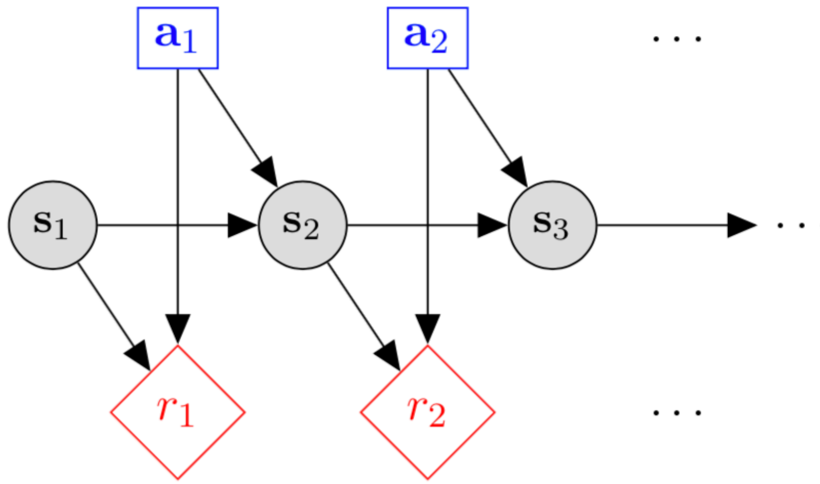
Policy: how to act in each state

Example: House cleaning robot



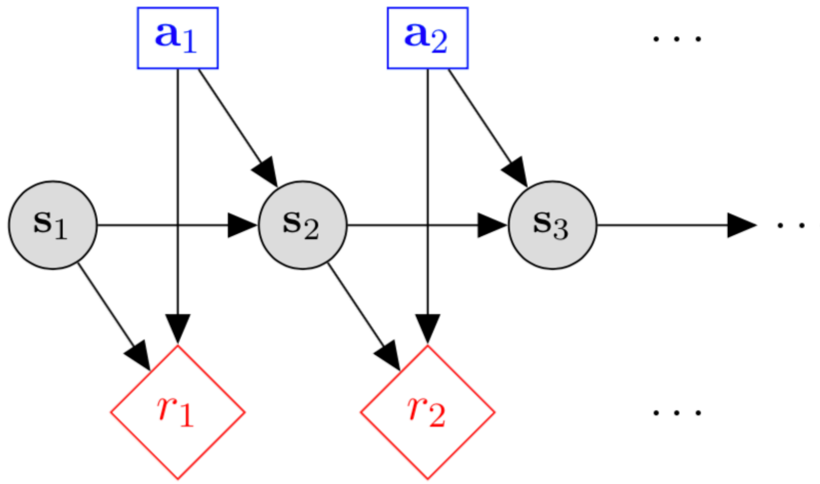
- States: physical location, objects in environment
- Actions: move, pick-up, drop, ...
- Reward: +1 if pick up dirty clothes, -1 if break dish, ...
- Transition model: describe actuators and uncertain environment

Example: Game of Go



- States: board position
- Actions: move a piece
- Reward: +1 if win the game, 0 if draw, -1 if lose the game
- Transition model: rules of game, response of other player

Example: Customer service bot



- States: summary of conversation so far
- Actions: words to utter
- Reward: +1 if solve caller's problem, -1 if need to go to human, -10 if caller hangs up angry
- Transition model: effect of words on state, next words or action from caller.

Two Kinds of Problems

- “Planning”: Given knowledge of the probabilistic model of the MDP, compute an optimal policy
- “Reinforcement learning”: Given access to the world (or a simulator of the world), learn an optimal policy