Topics of Active Research in Reinforcement Learning Relevant to Spoken Dialogue Systems

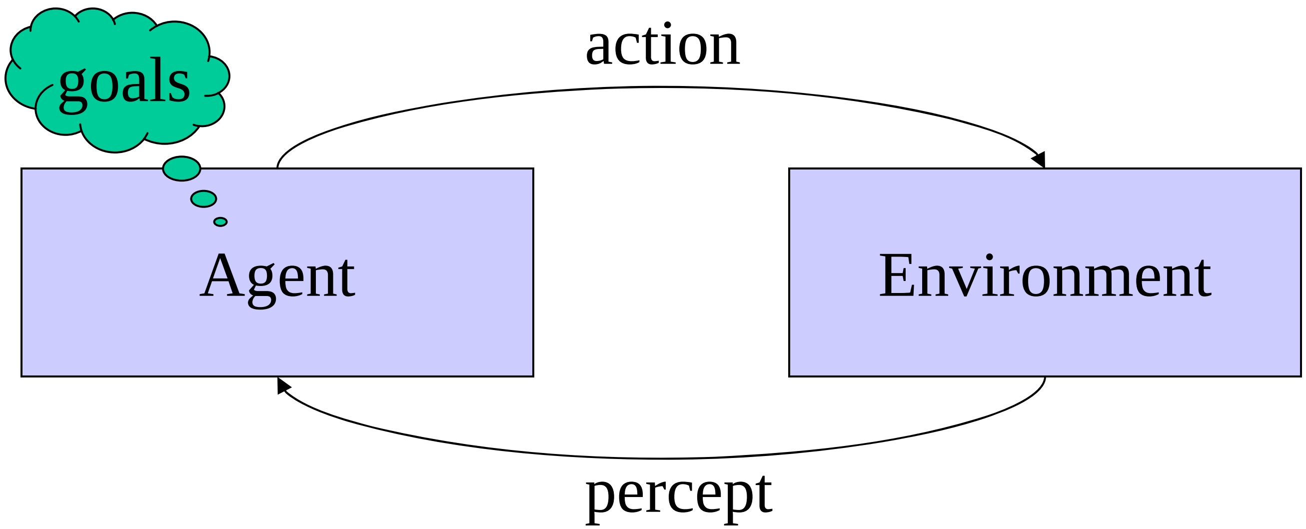
Pascal Poupart
David R. Cheriton School of Computer Science
University of Waterloo

Outline

- Review
  - Markov Models
  - Reinforcement Learning
- Some areas of active research relevant to SDS
  - Bayesian Reinforcement Learning (BRL)
  - Inverse Reinforcement Learning (IRL)
  - Predictive State Representations (PSRs)
- Conclusion

Automated System

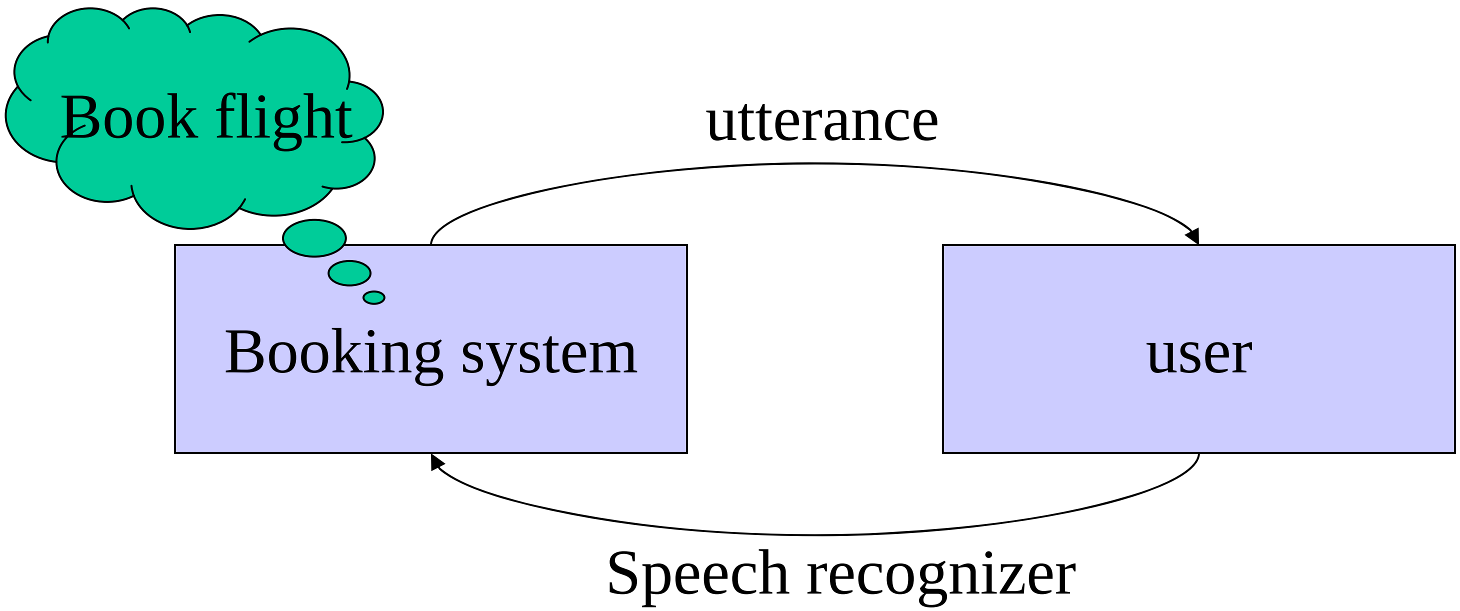
- Abstraction:



- Problems:
  - Uncertain action effects
  - Imprecise percepts
  - Unknown environment

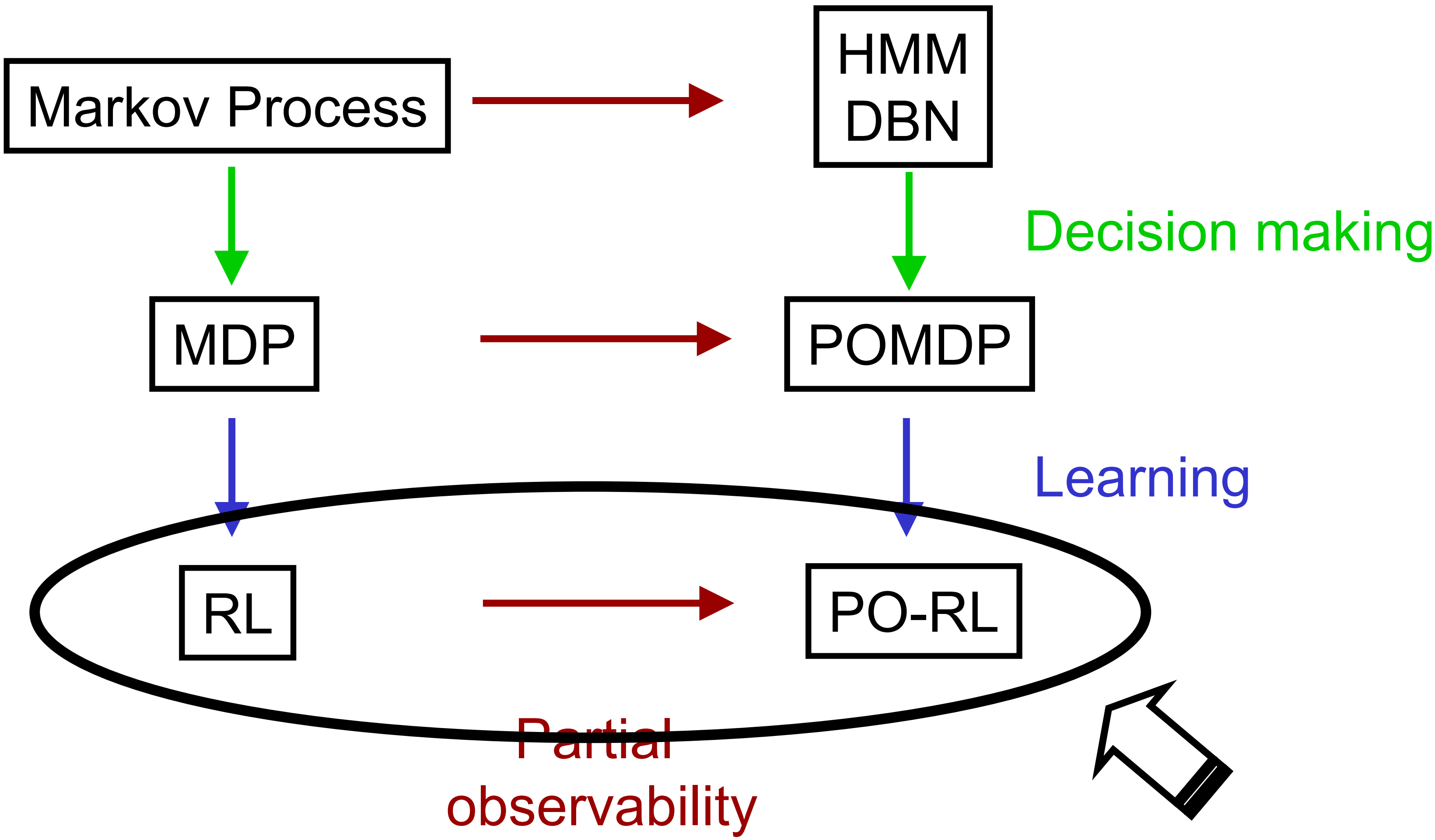
Automated System

- Spoken Dialogue System:



- Problems:
  - User may not hear/understand system utterances
  - Imprecise speech recognizer
  - Unknown user model

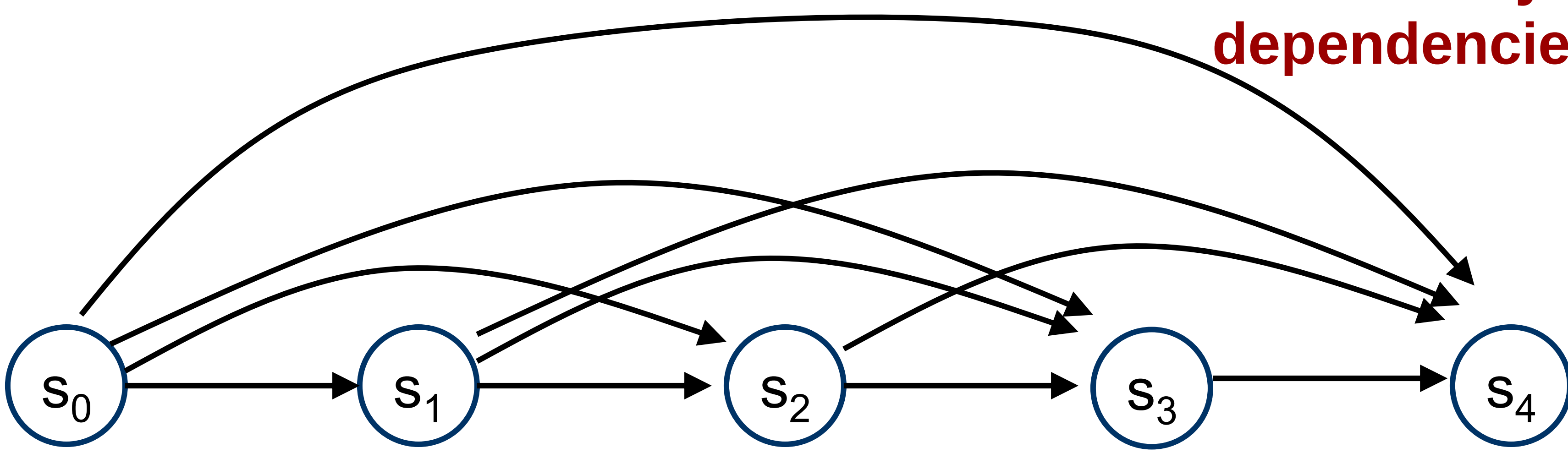
Markov Models



Stochastic Process

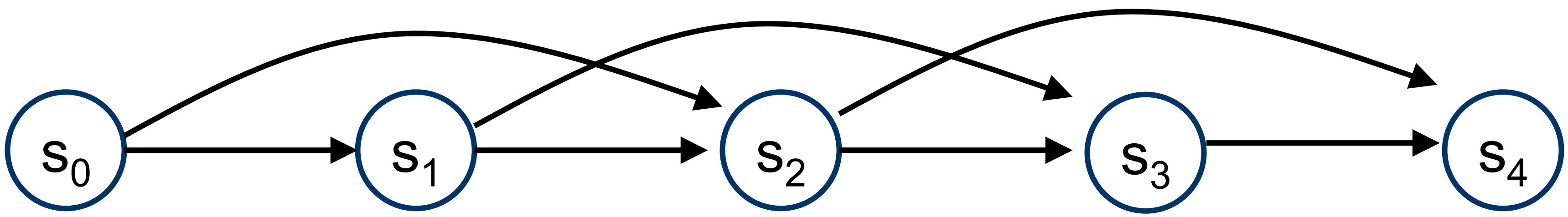
- World state changes over time
- Convention:
  - Circle: Random variable
  - Arc: Conditional dependency
    - Stochastic dynamics: $\Pr(s_{t+1} | s_t, \ldots, s_0)$

Too many dependencies!



Markov Process

- Markov assumption: current state depends only on finite history of past states
  - K-order Markov process: $Pr(s_t|s_{t-1},\ldots,s_{t-k},\cancel{s_{t-k-1},\ldots,s_0}) = Pr(s_t|s_{t-1},\ldots,s_{t-k})$

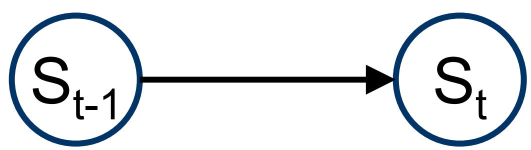


- Example:
  - N-gram model: $Pr(word_i|word_{i-1}, \ldots, word_{i-n})$

Markov Process

- Stationary Assumption: dynamics do not change
  - $Pr(s_t|s_{t-1},\ldots,s_{t-k})$ is same for all t

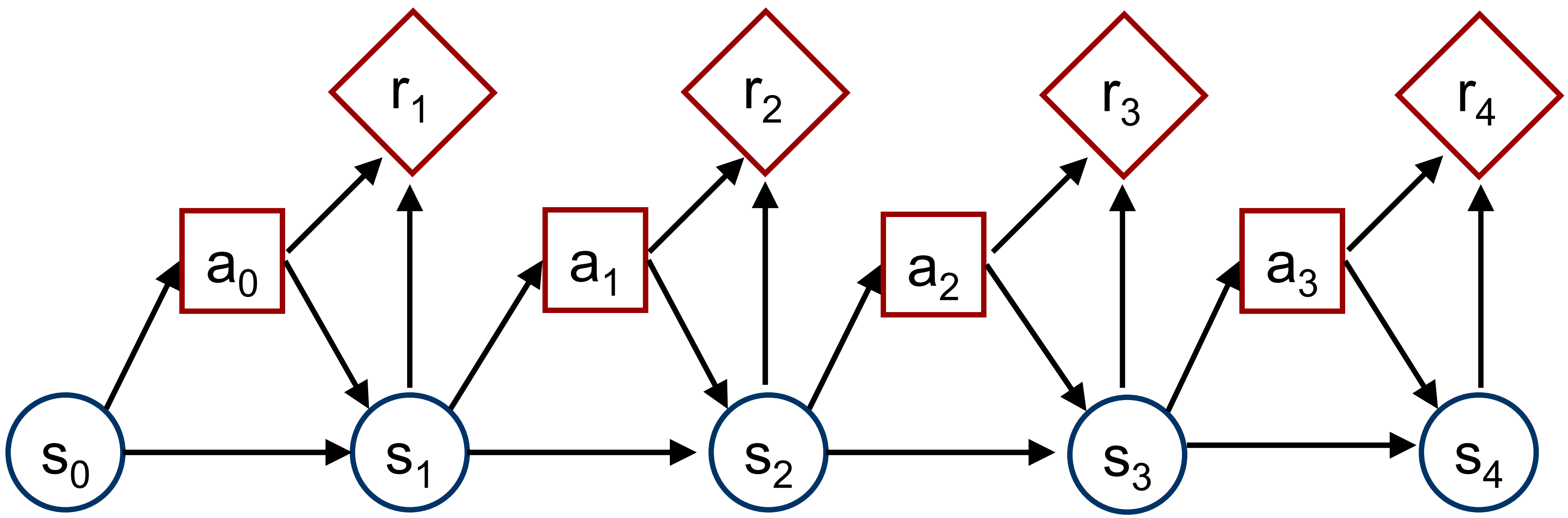
- Two slices sufficient for a first-order Markov process…
  - Graph:



  - Dynamics: $Pr(s_t|s_{t-1})$
  - Prior: $Pr(s_0)$

Markov Decision Process

- Intuition: (First-order) Markov Process with…
  - Decision nodes
  - Utility nodes



Markov Decision Process

- Definition
  - Set of states: S
  - Set of actions: A
  - Transition model: $T(s_{t-1},a_{t-1},s_t) = \Pr(s_t|a_{t-1},s_{t-1})$
  - Reward model: $R(s_t,a_t) = r_t$
  - Discount factor: $0 \leq \gamma \leq 1$

- Goal: find optimal policy
  - Policy π: $S \rightarrow A$
  - Value: $V^{\pi}(s) = E_{\pi} [\Sigma_t \gamma^t r_t]$
  - Optimal policy π^*: $V^{\pi^*}(s) \geq V^{\pi}(s) \; \forall \pi,s$

MDPs for SDS

- MDPs for SDS: Biermann and Long (1996), Levin and Pieraccini (1997), Singh et al. (1999), Levin et al. (2000)

- Flight booking example:
  - State: Assignment of values to dep. date, dep. time, dep. city and dest. city
  - Actions: any utterance (e.g., question, confirmation)
  - User model: Pr(user response | sys. utterance, state)
  - Rewards: positive reward for correct booking, negative reward for incorrect booking

Value Iteration

- Three families of algorithms:
  - Value iteration, policy iteration, linear programming

- Bellman's equation:
  - $V(s_t) = \max_{a_t} R(s_t,a_t) + \gamma \sum_{s_{t+1}} \Pr(s_{t+1}|s_t,a_t) V(s_{t+1})$
  - $a_t^* = \mathrm{argmax}_{a_t} R(s_t,a_t) + \gamma \sum_{s_{t+1}} \Pr(s_{t+1}|s_t,a_t) V(s_{t+1})$

- Value iteration:
  - $V(s_h) = R(s_h)$
  - $V(s_{h-1}) = \max_{a_{h-1}} R(s_{h-1},a_{h-1}) + \gamma\sum_{s_h}\Pr(s_h|s_{h-1},a_{h-1})V(s_h)$
  - $V(s_{h-2}) = \max_{a_{h-2}} R(s_{h-2},a_{h-2}) + \gamma\sum_{s_{h-1}}\Pr(s_{h-1}|s_{h-2},a_{h-2})V(s_{h-1})$

Unrealistic Assumptions

- Transition (user) model known:
  – How to learn a good user model?
- Reward model known:
  – How to assess user preferences?
- Speech recognizer flawless:
  – How to account for ASR errors?

RL

IRL

PORL

POMDP

Reinforcement Learning

- Markov Decision Process:
  - **S:** set of states
  - **A:** set of actions
  - **R(s,a) = r:** reward model
  - **T(s,a,s') = Pr(s'|s,a):** transition function

Reinforcement Learning

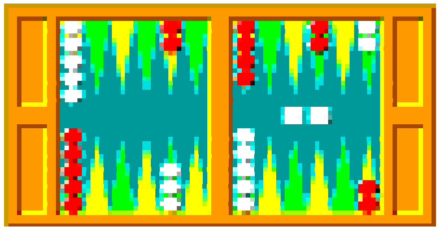
- RL for SDS: Walker et al. (1998), Singh et al. (1999), Scheffler and Yound (1999), Litman et al. (2000), Levin et al. (2000), Pietquin (2004), Georgila et al (2005), Lewis & Di Fabbrizio (2006)

Algorithms for RL

- Model-based RL:
  - Estimate T from s,a,s' triples
    - E.g., Max likelihood: Pr(s'|s,a) = #(s,a,s') / #(s,a,•)
  - Model learning: offline (corpus of s,a,s' triples) and/or
    online (s,a,s' directly from env.)

- Model-free RL:
  - Estimate V^* and/or π^* directly
    - E.g., Temporal difference:
      $Q(s,a) = Q(s,a) + \alpha\,[R(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a)]$
  - Learning: offline (s,a,s' from simulator)
    online (s,a,s' directly from environment)

Successes of RL



- Backgammon [Tesauro 1995]
  - Temporal difference learning
  - Trained by self-play
  - Simulator: opponent model consists of itself
  - Offline learning: simulated millions of games

- Helicopter control [Ng et al. 2003,2004]
  - PEGASUS: stochastic gradient descent
  - Offline learning: with flight simulator

Outline

- Review
  - Markov Models
  - Reinforcement Learning
- Some areas of active research relevant to SDS
  - Bayesian Reinforcement Learning (BRL)
  - Inverse Reinforcement Learning (IRL)
  - Predictive State Representations (PSRs)
- Conclusion

Assistive Technologies

- Handwashing assistant
  - [Boger et al. IJCAI-05]

- Use RL to adapt to users
  - Start with basic user model
  - Online learning:
    - Adjust model as system interacts with users
    - Bear cost of actions
    - Cannot explore too much
    - Real-time response

Bayesian Model-based RL

- Formalized in Operations Research by Howard and his students at MIT in the 1960s
- In AI: Kaelbling (1992), Meuleau and Bourgine (1999), Dearden & al (1998,1999), Strens (2000), Duff (2003), Wang & al (2005), Poupart & al (2006)

- Advantages
  - Opt. exploration/exploitation tradeoff
  - Encode prior knowledge

less data required

Bayesian Model-based RL

- Disadvantage:
  - Computationally complex
- Poupart et al. (ICML 2006):
  - **Optimal value function has simple parameterization**
    - i.e., upper envelope of a set of multivariate polynomials
  - **BEETLE: Bayesian Exploration/Exploitation Tradeoff in LEarning**
    - Exploit polynomial parameterization

Bayesian RL

- Basic Idea:
- Encode unknown prob. by random variables θ
  - i.e., $\theta_{sas'} = \Pr(s'|s,a)$: random variable in [0,1]
  - i.e., $\theta_{sa} = \Pr(\bullet|s,a)$: multinomial distribution

- Model learning: update $\Pr(\theta)$
- $\Pr(\theta)$ tells us which part of the model are not well known and therefore worth exploring

Model Learning

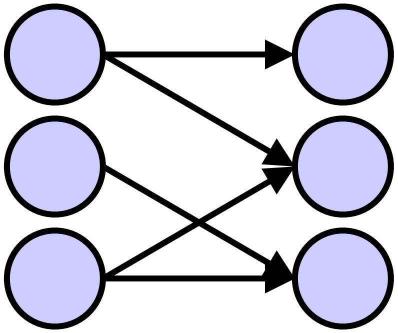
- Assume prior $b(\theta_{sa}) = Pr(\theta_{sa})$
- Learning: compute posterior given s,a,s'
  - $b_{sas'}(\theta_{sa}) = k\ Pr(\theta_{sa})\ Pr(s'|s,a,\theta_{sa}) = k\ b(\theta_{sa})\ \theta_{sas'}$
- **Conjugate prior:**
  - Dirichlet prior → Dirichlet posterior
- $b(\theta_{sa}) = Dir(\theta_{sa};n_{sa}) = k\ \Pi_{s''}\ (\theta_{sas''})^{n_{sas''} - 1}$
- $b_{sas'}(\theta_{sa}) = k\ b(\theta_{sa})\ \theta_{sas'}$

$$= k\ \Pi_{s''}\ (\theta_{sas''})^{n_{sas''} - 1 + \delta(s',s'')}$$

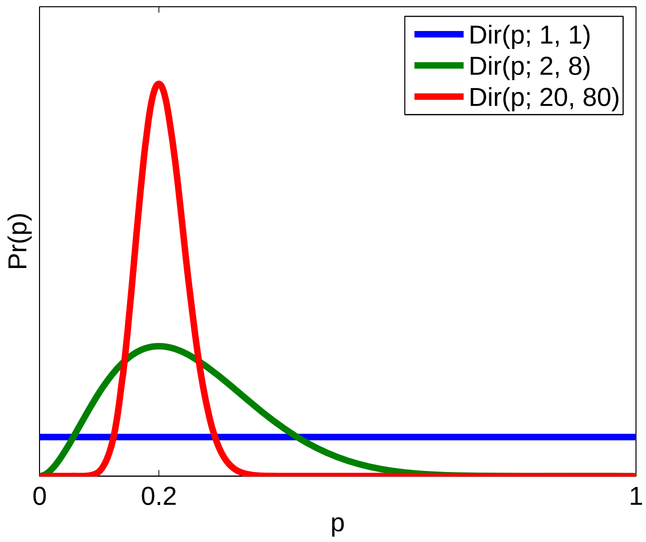
$$= k\ Dir(\theta_{sa};\ n_{sa} + \delta(s',s''))$$

Prior Knowledge

- Structural priors
  - Tie identical parameters
    - If Pr(•|s,a) = Pr(•|s',a') then $\theta_{sa} = \theta_{s'a'}$
  - Factored representation
    - DBN: unknown conditional dist.



- Informative priors
  - No knowledge: uniform Dirichlet
  - If $(\theta_1, \theta_2) \sim (0.2, 0.8)$ then set (n_1, n_2) to $(0.2k, 0.8k)$
    - k indicates the level of confidence



Policy Optimization

- Classic RL:
  - **$V^*(s) = \max_a R(s,a) + \Sigma_{s'} \Pr(s'|s,a)\, V^*(s')$**
  - Hard to tell what needs to be explored
  - Exploration heuristics: ε-greedy, Boltzmann, etc.

- Bayesian RL:
  - **$V^*(s,b) = \max_a R(s,a) + \Sigma_{s'} \Pr(s'|s,b,a)\, V^*(s',b_{sas'})$**
  - Belief b tells us what parts of the model are not well known and therefore worth exploring
  - Optimal exploration/exploitation tradeoff

Value Function Parameterization

- **Theorem:** V^* is the upper envelope of a set of multivariate polynomials ($V_s(\theta) = \max_i poly_i(\theta)$)
- **Proof:** by induction
  – Define value function in terms of θ instead of b
    - i.e. $V^*(s,b) = \int_\theta b(\theta)\, V_s(\theta)\, d\theta$

  – Bellman's equation
    - $$\begin{aligned} V_s(\theta) &= \max_a R(s,a) + \Sigma_{s'} Pr(s'|s,a,\theta)\ V_{s'}(\theta) \\ &= \max_a \underbrace{k_a + \Sigma_{s'}\ \theta_{sas'}\ \max_i poly_i(\theta)}_{poly_j(\theta)} \\ &= \max_j\ poly_j(\theta) \end{aligned}$$

BEETLE Algorithm

- Sample a set of reachable belief points B
- $V \leftarrow \{0\}$
- Repeat
  - $V' \leftarrow \{\}$
  - For each b in B compute multivariate polynomial
    - $poly_{as'}(\theta) \leftarrow \text{argmax}_{poly \in V} \int_\theta b_{sas'}(\theta)\, poly(\theta)\, d\theta$
    - $a^* \leftarrow \text{argmax}_a \int_\theta b_{sas'}(\theta)\, R(s,a) + \Sigma_{s'}\, \theta_{sas'}\, poly_{as'}(\theta)\, d\theta$
    - $poly(\theta) \leftarrow R(s,a^*) + \Sigma_{s'}\, \theta_{sa^*s'}\, poly_{a^*s'}(\theta)$
    - $V' \leftarrow V' \cup \{poly\}$
  - $V \leftarrow V'$

Bayesian RL

- Summary:
  - Optimizes exploration/exploitation tradeoff
  - Easily encode prior knowledge to reduce exploration
- Potential for SDS:
  - Online user modeling:
    - Tailor model to specific user with least exploration possible
  - Offline user modelling:
    - Large corpus of unlabeled dialogues
    - Labeling takes time
    - Automated selection of a subset of dialogues to be labeled
    - Active learning: Jaulmes, Pineau et al. (2005)

Outline

- Review
  – Markov Models
  – Reinforcement Learning

- Some areas of active research relevant to SDS
  – Bayesian Reinforcement Learning (BRL)
  – Inverse Reinforcement Learning (IRL)
  – Predictive State Representations (PSRs)

- Conclusion

Reward Function

- MDPs: T and R → π
- RL: s,a,s' and R → π
- But R is often difficult to specify!

- SDS booking system:
  - Correct booking: large positive reward
  - Incorrect booking: large negative reward
  - Cost per question: ???
  - Cost per confirmation: ???
  - User frustration: ???

Apprenticeship learning

- Sometimes: expert policy π^+ observable

- Apprenticeship learning:
  – Imitation: $\pi^+ \rightarrow \pi$
    - When T doen't change, just imitate policy directly
  – Inverse RL: π^+ and s,a,s' $\rightarrow$ R
    - When T could change, estimate R
    - Then do RL: s,a,s' and R $\rightarrow \pi$
    - For different SDS, we have different policies because of different scenarios, but perhaps the same R can be used.

Inverse RL

- In AI: Ng and Russell (2000), Abbeel and Ng (2004), Ramachandran and Amir (2006)

- Bellman's equation:
  - $V(s_t) = \max_{a_t} R(s_t,a_t) + \gamma \Sigma_{s_{t+1}} \Pr(s_{t+1}|s_t,a_t)\, V(s_{t+1})$

- Idea: find R such that π^+ is optimal according to Bellman's equation.
- Bayesian Inverse RL:
  - Prior Pr(R)
  - Posterior Pr(R|s,a,s',a',s'',a'',…)

Outline

- Review
  - Markov Models
  - Reinforcement Learning
- Some areas of active research relevant to SDS
  - Bayesian Reinforcement Learning (BRL)
  - Inverse Reinforcement Learning (IRL)
  - Predictive State Representations (PSRs)
- Conclusion

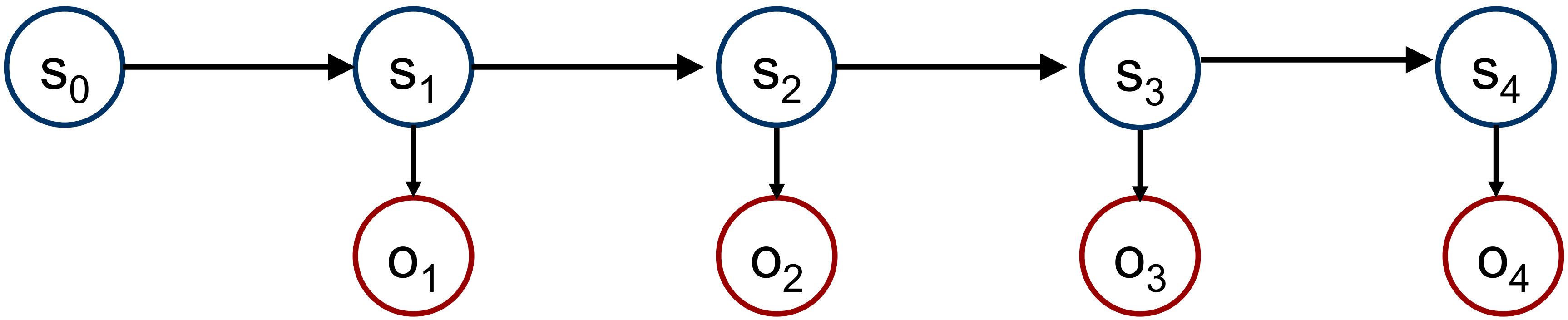
Partially Observable RL

- States are rarely observable
- Noisy sensors: measurements are correlated with states of the world
- Extend Markov models to account for sensor noise

- Recall:
  - Markov Process → HMM
  - MDP → POMDP
  - RL → PORL

Hidden Markov Model

- Intuition: Markov Process with …
  - Observation variables



- Example: speech recognition

Hidden Markov Model

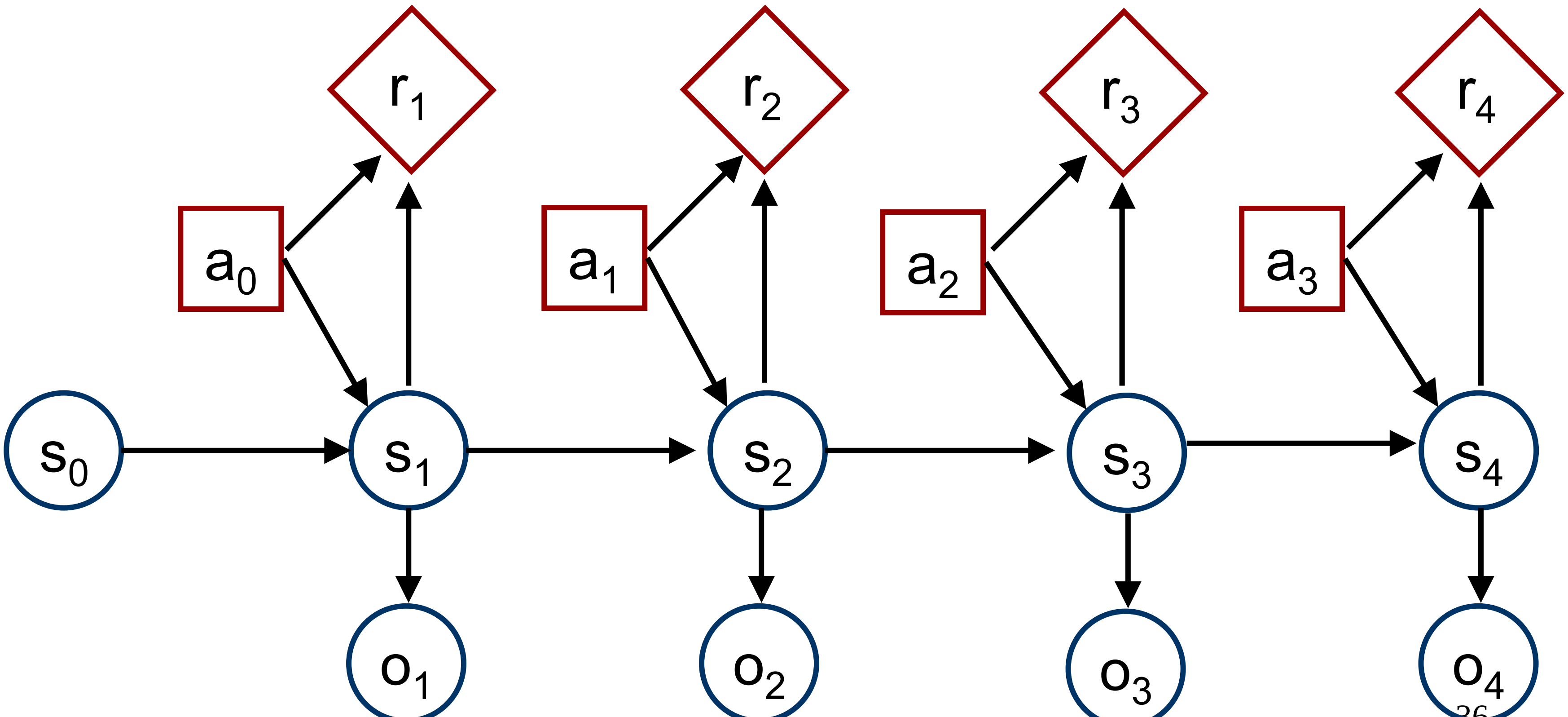
- Definition
  - Set of states: S
  - Set of observations: O
  - Transition model: $Pr(s_t|s_{t-1})$
  - Observation model: $Pr(o_t|s_t)$
  - Prior: $Pr(s_0)$

- Belief monitoring:
  - Prior: $b(s) = Pr(s)$
  - Posterior: $b_{ao}(s') = Pr(s'|a,o)$

$$= k \Sigma_s b(s) Pr(s'|s,a) Pr(o|s')$$

Partially Observable MDP

- Intuition: HMM with…
  - Decision nodes
  - Utility nodes



Partially Observable MDP

- Definition
  - Set of actions: A
  - Set of observations: O
  - Reward model: $R(s_t,a_t) = r_t$
  - Set of states: S
  - Transition model: $T(s_{t-1},a_{t-1},s_t) = Pr(s_t|a_{t-1},s_{t-1})$
  - Observation model: $Z(s_t,o_t) = Pr(o_t|s_t)$

- POMDPs for SDS: Roy et al. (2000), Zhang et al. (2001), Williams et al. (2006), Atrash and Pineau (2006)

Partially Observable RL

- Definition
  - Set of actions: A
  - Set of observations: O
  - Reward model: $R(s_t,a_t) = r_t$
  - Set of states: S
  - Transition model: $T(s_{t-1},a_{t-1},s_t) = Pr(s_t|a_{t-1},s_{t-1})$
  - Observation model: $Z(s_t,o_t) = Pr(o_t|s_t)$

PO-RL

- NB: S is generally unknown since it is an unobservable quantity

PORL algorithms

- Model-free PORL:
  - Stochastic gradient descent
- Model-based PORL:
  - Assume S, learn T and Z from a,o,a',o',… sequences
    - E.g. EM algorithm for HMMs
    - But S is really unknown
    - In SDS, S may refer to user intentions, mental state, language knowledge, etc.
  - Learn S, T and Z from a,o,a',o',… sequences
    - E.g., Predictive state representations

Sufficient statistics

- Beliefs are sufficient statistics to predict future observations
  - $\Pr(o|b) = \Sigma_s\, b(s) \Pr(o|s)$
  - $\Pr(o'|b,o,a) = k\, \Sigma_s\, b(s) \Pr(o|s)\, \Sigma_{s'} \Pr(s'|s,a) \Pr(o'|s')$
    $= \Sigma_{s'}\, b_{o,a}(s') \Pr(o'|b_{o,a})$
  - …
- Are there more succinct sufficient statistics?

Predictive State Representations

- Belief b
  - vector of probabilities
  - Information to predict future observations
  - After each o,a pair, b is updated to $b_{o,a}$ using T and Z

- Idea: find sufficient statistic x such that
  - x is a vector of real numbers
  - x is a smaller vector than b
  - There exist functions f and g such that
    - $f(x) = Pr(o|b)$
    - $g(x,a,o) = x_{o,a}$ and $f(g(x,a,o)) = Pr(o|b,o,a)$

Predictive State Representations

- References: Litman et al. (2002), Poupart and Boutilier (2003), Singh et al. (2003), Rudary and Singh (2004), James and Singh (2004), etc.

- Potential for SDS
  - Instead of handcrafting state variables, learn a state representation of users from data
  - Learn a smaller user model

Conclusion

- Overview of Markov models for SDS
- RL topics of active research relevant to SDS:
  - Bayesian RL: tailor model to specific users
  - Inverse RL: learn reward model
  - PSR: learn state representation of user model

- Fields of machine learning and user modelling could offer more techniques to advance SDS