Reinforcement Learning of POMDPs using Spectral Methods

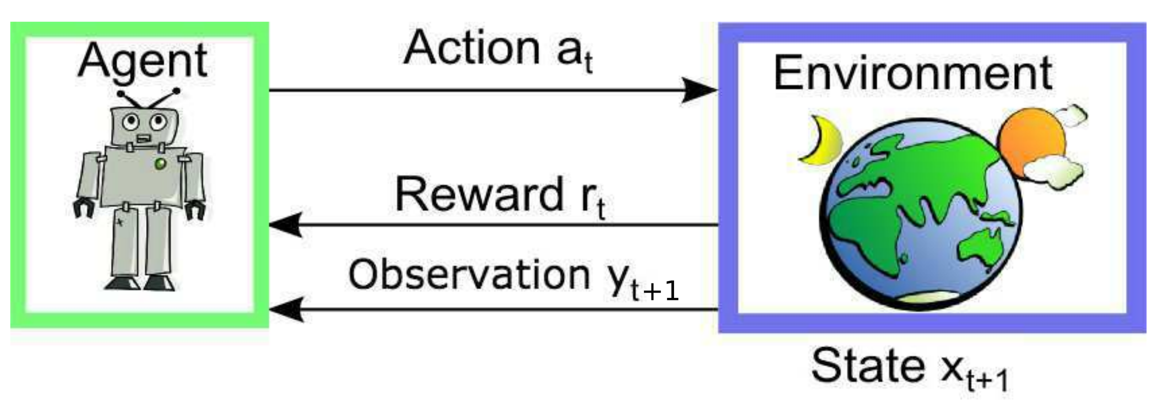
Kamyar Azizzadenesheli* Alessandro Lazaric† Animashree Anandkumar*

*University of California, Irvine (UCI) †Institut National de Recherche en Informatique et en Automatique, (Inria)

Reinforcement Learning

Learning in Adaptive Environments:

- Environment-Agent Interaction.
- Reinforcement Learning: feedback or rewards to reinforce policy.
- History: $\mathcal{H} := \{y_1, a_1, r_1, \dots, a_{t-1}, r_{t-1}, y_t\}$
- Policy is a mapping $\pi : \mathcal{H} \rightarrow \mathcal{A}$.
- No prior knowledge
  - Learning (Exploring)
  - Planning (Exploiting)
- Objective: $\max_{\pi} \eta_\pi = \sum_t r_t$



Partially Observable Models

POMDP vs MDP

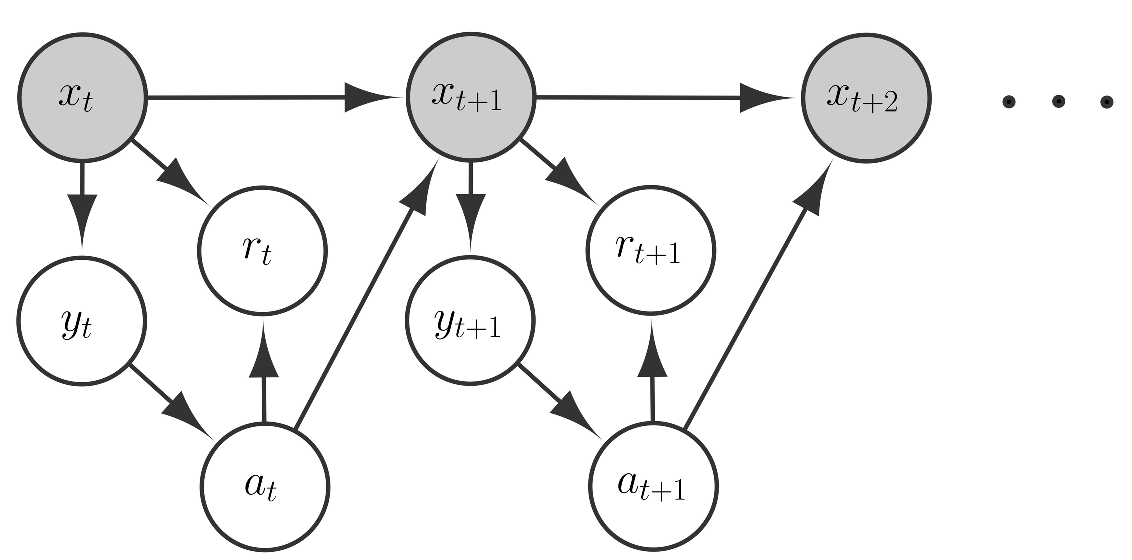
- Captures the hidden structures
- Captures the latent factors(unobservable effects)
- No Markovian assumption on observation level

Disadvantages

- Hard Learning and Planning

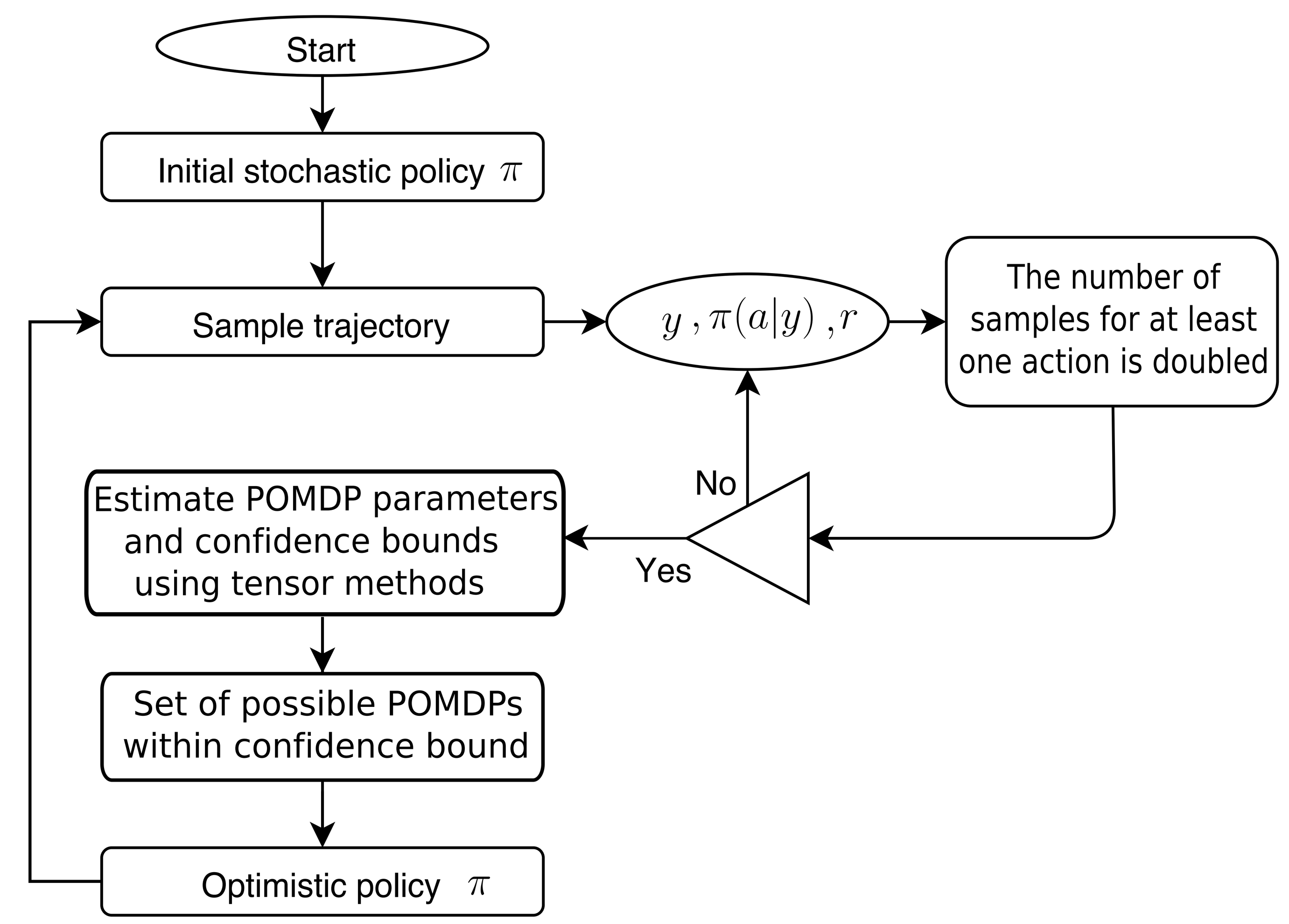
POMDP graphical model:

$$\text{Parameters of interest} = \begin{cases} T_{x',x,a} = \mathbb{P}(X' = x' | X = x, A = a) \\ O_{y,x} = \mathbb{P}(Y = y | X = x) \\ \Gamma_{r,a,x} = \mathbb{P}(R = r | X = x, A = a) \end{cases}$$



SM-UCRL-POMDP

- Apply policy π until the number of samples, at least for one action is doubled
- Compute the plausible set of models → Find the optimal policy w.r.t optimistic model



Spectral Methods

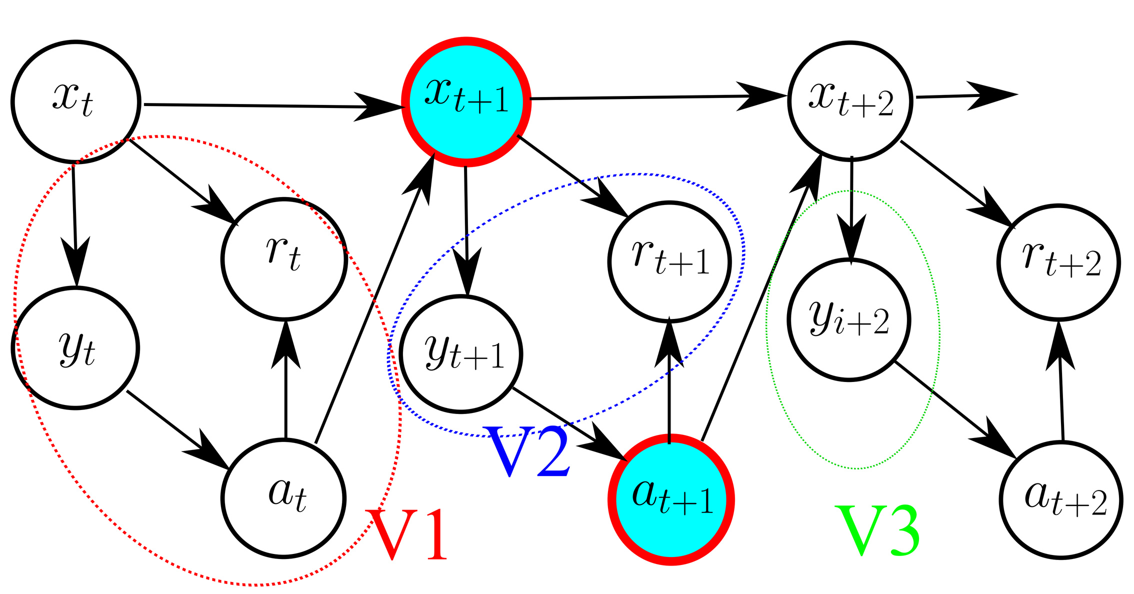
Tensor Decomposition:
Multiview model condition on middle action and middle state

Tensor Moments

- $v_t \perp v_{t+1} \perp v_{i+2} | x_{t+1}, a_{t+1}$
- $V_1^{(l)} = \mathbb{P}\left(\vec{y}_1, \vec{r}_1, a_1 | x_2, a_2 = l\right)$,
- $V_2^{(l)} = \mathbb{P}\left(\vec{y}_2, \vec{r}_2 | x_2, a_2 = l\right)$,
- $V_3^{(l)} = \mathbb{P}\left(\vec{y}_3 | x_2, a_2 = l\right)$.

$$\boxed{\mathbb{E}[v_1 \otimes v_2 \otimes v_3 | a_2 = l] = \textstyle\sum_j \omega_\pi^{(l)} \cdot [V_1^{(l)}]_{:,j} \otimes [V_2^{(l)}]_{:,j} \otimes [V_3^{(l)}]_{:,j}.}$$

Multiview Model



Parameter Learning

Second and Third order moments given middle action **Confidence intervals**

$$\left.\begin{array}{l} M_2^{(l)} = \sum_x \omega^{(l)}(x)[V_1^{(l)}]_{:,x} \otimes [V_3^{(l)}]_{:,x} \\ M_3^{(l)} = \sum_x \omega^{(l)}(x)[V_1^{(l)}]_{:,x} \otimes [V_3^{(l)}]_{:,x} \otimes [V_2^{(l)}]_{:,x} \end{array}\right\} \Rightarrow \begin{array}{l} \|\widehat{O}(:,i) - O(:,i)\|_1 = \mathcal{O}\left(\sqrt{\frac{Y \log(1/\delta)}{T_l}}\right), \\ \|\widehat{T}(\cdot|i,l) - T(\cdot|i,l)\|_1 = \mathcal{O}\left(\sqrt{\frac{Y \cdot X^2 \log(1/\delta)}{T_l}}\right). \end{array}$$

Regret Analysis

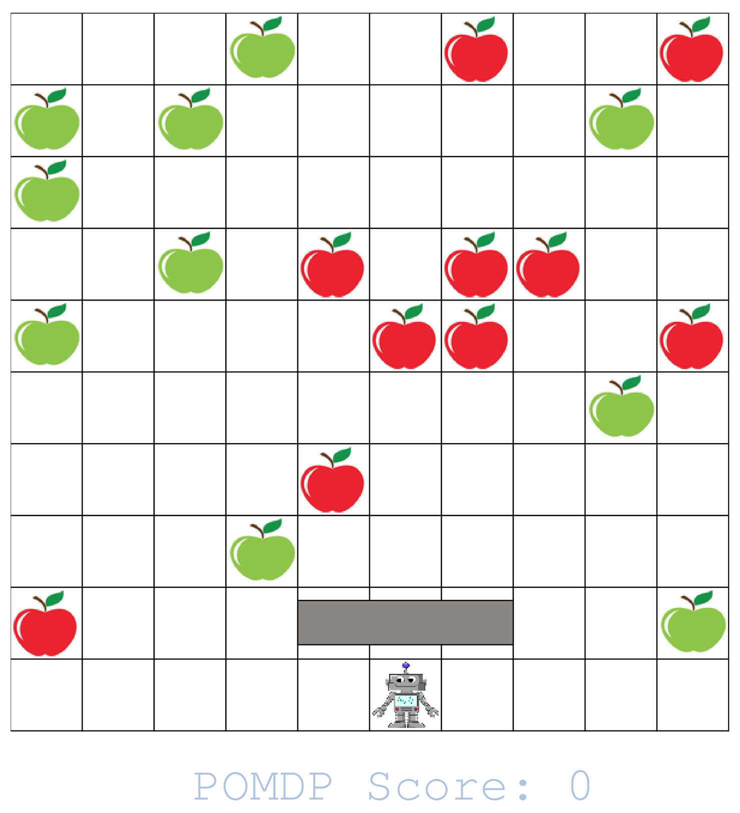
$$\boxed{POMDPs;\ \text{Regret}(T) = \tilde{\mathcal{O}}\left(DX\sqrt{A \cdot Y \cdot X \cdot T}\right)}$$

Extended results on Regret Analysis of CMDPs

$$\boxed{CMDPs;\ \text{Regret}(T) = \tilde{\mathcal{O}}\left(D_{MDP}X\sqrt{A \cdot T}\right)}$$

Experimental results

Grid world



Experimental results

Game setting

- Rewards metric: green apple = +1, red apple = -1
- The apples are randomly generated and removed
- Partially observed environment, 3 boxes visible

SM-UCRL-POMDP vs DQN

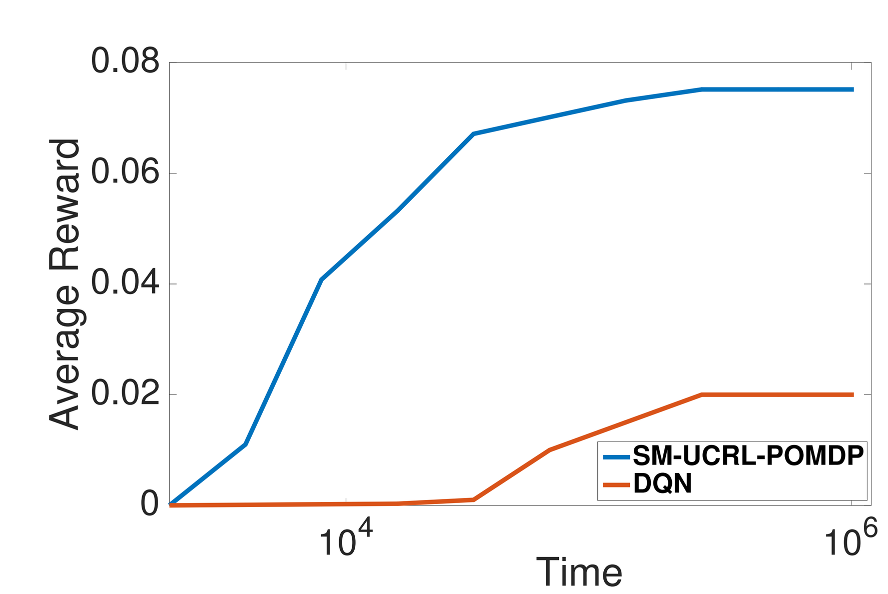
- SM-UCRL-POMDP → tuned by 8 hidden states.
- DQN → 3 hidden layers, 30 hyperbolic tangent unites at each layer with RMSprop update

RMSProp

- $r_t = (1-\gamma) f'(\theta_t)^2 + \gamma r_{t-1}$,
- $v_{t+1} = \frac{\alpha}{\sqrt{r_t}} f'(\theta_t)$,
- $\theta_{t+1} = \theta_t - v_{t+1}$.

Conclusion

- Model misspecification
- Regret
- Robustness
- Convergence
- Sample complexity and Computation cost (Seconds VS Hours)



Email: kazizzad@uci.edu