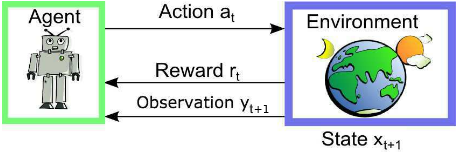
Reinforcement Learning of POMDPs using Tensor Methods

Kamyar Azizzadenesheli

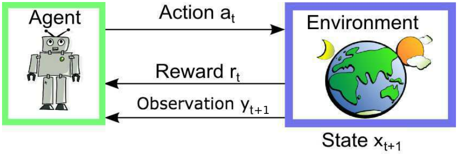
U.C. Irvine

Joint work with Prof. Anima Anandkumar and Dr. Alessandro Lazaric.

Learning in Adaptive Environments



Learning in Adaptive Environments



- Environment-Agent Interaction.
- History: $\mathcal{H} := \{y_1, a_1, r_1, \ldots, a_{t-1}, r_{t-1}, y_t\}$

- Reinforcement Learning: feedback or rewards to reinforce policy.
- Policy is a mapping $\pi : \mathcal{H} \rightarrow \mathcal{A}$.

Model-based Reinforcement Learning

Agent-Environment Interaction

- Policy $\mathbb{P}(a_t|y_t, r_{t-1}, \ldots, y_1)$.
- Reward Probability: $\mathbb{P}(r_t|a_t, y_t, \ldots, y_1)$.
- Transition Probability: $\mathbb{P}(y_{t+1}|r_t, a_t, y_t, \ldots, y_1)$.

No prior knowledge

- Learning (Exploring).
- Planning (Exploiting).

Efficient modeling frameworks?

Markovian Processes

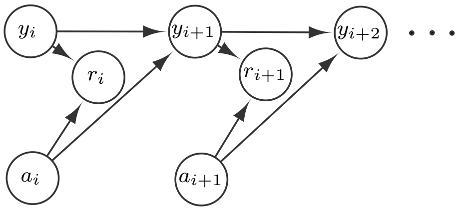
Markov Decision Process (MDP)

- Fully Observable Environment: $y_t = x_t, \ \forall t \in \{1, \ldots, T\}$.
- Markovian Assumption:
  - $\mathbb{P}(y_{t+1}|r_t, a_t, y_t, r_{t-1}, \ldots, y_1) = \mathbb{P}(y_{t+1}|a_t, y_t)$.
  - $\mathbb{P}(r_t|a_t, y_t, r_{t-1}, a_{t-1}, \ldots, y_1) = \mathbb{P}(r_t|a_t, y_t)$.

Markovian Processes

Markov Decision Process (MDP)

- Fully Observable Environment: $y_t = x_t, \ \forall t \in \{1, \ldots, T\}$.
- Markovian Assumption:
  - $\mathbb{P}(y_{t+1}|r_t, a_t, y_t, r_{t-1}, \ldots, y_1) = \mathbb{P}(y_{t+1}|a_t, y_t)$.
  - $\mathbb{P}(r_t|a_t, y_t, r_{t-1}, a_{t-1}, \ldots, y_1) = \mathbb{P}(r_t|a_t, y_t)$.



Markovian Processes

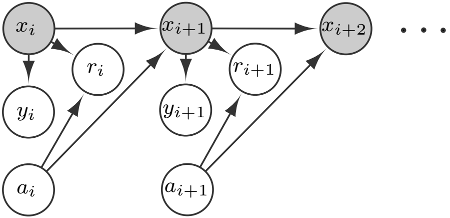
Partially Observable Markov Decision Process (POMDP)

- Evolution of hidden state $x_t \rightarrow \mathbb{P}(x_{t+1}|a_t, x_t)$
- Reward $r_t \rightarrow \mathbb{P}(r_t|a_t, x_t)$
- Observation y_t.
  - $\mathbb{P}(y_t|x_t, y_{t-1}, x_{t-1} \ldots) = \mathbb{P}(y_t|x_t)$.

Markovian Processes

Partially Observable Markov Decision Process (POMDP)

- Evolution of hidden state $x_t \rightarrow \mathbb{P}(x_{t+1}|a_t, x_t)$
- Reward $r_t \rightarrow \mathbb{P}(r_t|a_t, x_t)$
- Observation y_t.
  - $\mathbb{P}(y_t|x_t, y_{t-1}, x_{t-1} \ldots) = \mathbb{P}(y_t|x_t)$.



Reinforcement Learning under POMDPs?

Challenges and our Results

Challenges

- Hard Learning in general POMDPs $\rightarrow$ Active Dynamic Hidden Structure
- Hard Planning $\rightarrow$ PSpace-Complete

Challenges and our Results

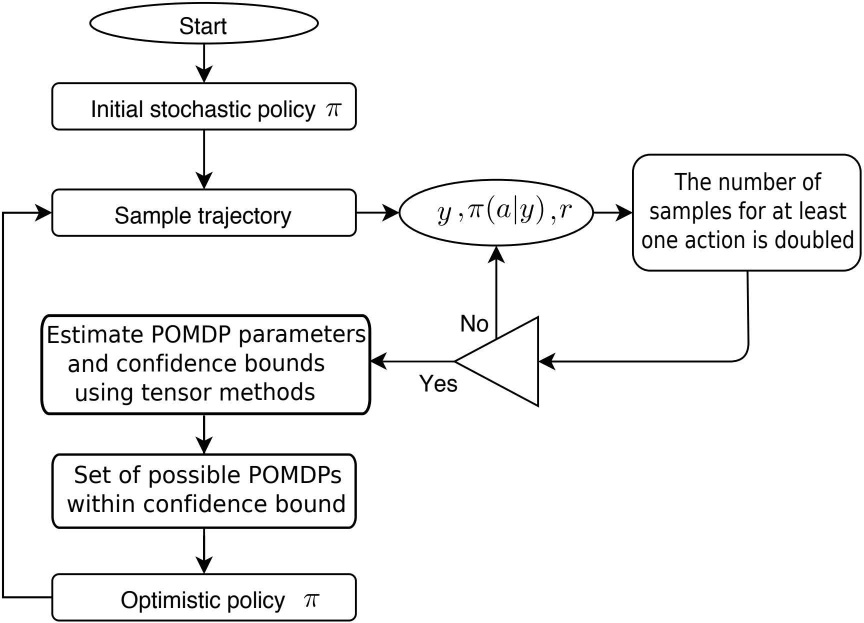
Challenges

- Hard Learning in general POMDPs $\rightarrow$ Active Dynamic Hidden Structure
- Hard Planning $\rightarrow$ PSpace-Complete

Our results RL POMDPs

- Novel learning algorithm with tensor decomposition methods
- Episodic learning and planning: Upper Confidence Reinforcement Learning (UCRL)
- Access to Oracle for Planning $\rightarrow$ $\widetilde{\mathcal{O}}(\sqrt{T})$ regret bound on memoryless setting

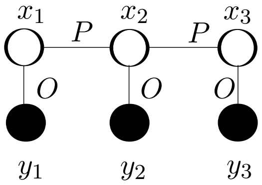
SM-UCRL-POMDP



Outline

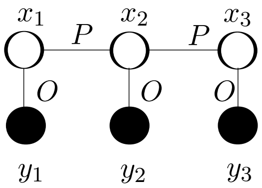
Warm-up: Learning HMMs

- O: Emission Matrix
- P: Transition Matrix



Warm-up: Learning HMMs

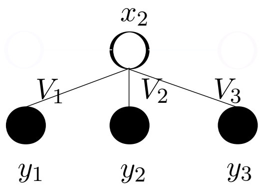
- O: Emission Matrix
- P: Transition Matrix



$\mathcal{CI}$

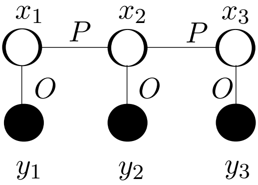
- $V_1 = \mathbb{E}[y_1|x_2]$
- $V_2 = \mathbb{E}[y_2|x_2] = O$
- $V_3 = \mathbb{E}[y_3|x_2] = OP$

$$\mathbb{E}[y_1 \otimes y_2 \otimes y_3] = \sum_i \omega_i \cdot V_{1_i} \otimes V_{2_i} \otimes V_{3_i}$$



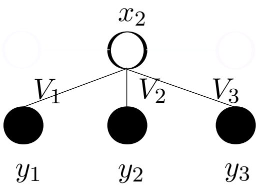
Warm-up: Learning HMMs

- O: Emission Matrix
- P: Transition Matrix



$\mathcal{CI}$

- $V_1 = \mathbb{E}[y_1|x_2]$
- $V_2 = \mathbb{E}[y_2|x_2] = O$
- $V_3 = \mathbb{E}[y_3|x_2] = OP$



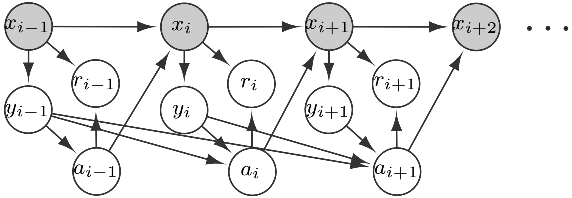
$$\mathbb{E}[y_1 \otimes y_2 \otimes y_3] = \sum_i \omega_i \cdot V_{1_i} \otimes V_{2_i} \otimes V_{3_i}$$

Conditions for Recovery

- Full column rank for observation matrix $O \in \mathbb{R}^{Y \times X}$ and P
- Ergodicity: ω and $P\omega$ have positive entries

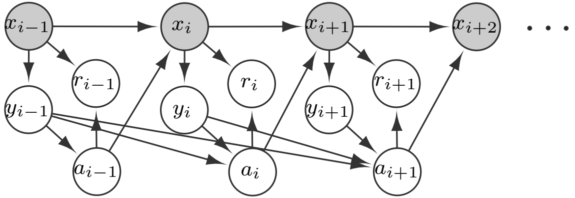
Challenges in Learning of POMDPs

Graphical model of a general POMDP

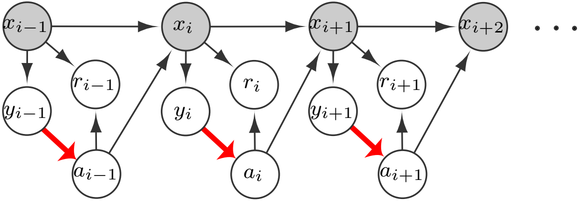


Challenges in Learning of POMDPs

Graphical model of a general POMDP

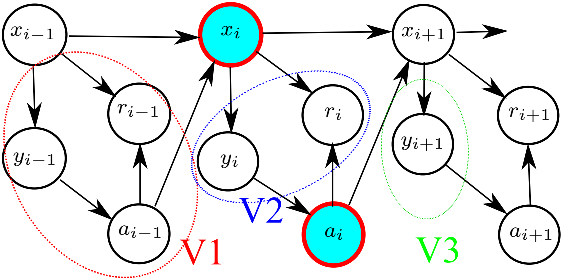


Simplification: limit to memoryless policies



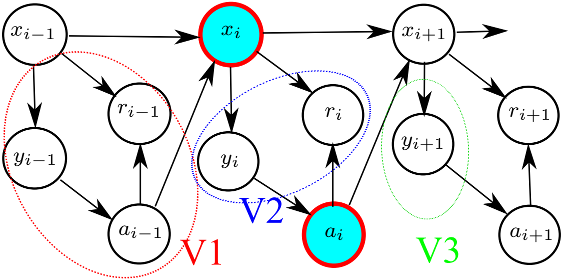
Learning POMDPs Under Fixed Memoryless Policies

- Fixed memoryless policy π throughout learning process.



Learning POMDPs Under Fixed Memoryless Policies

- Fixed memoryless policy π throughout learning process.

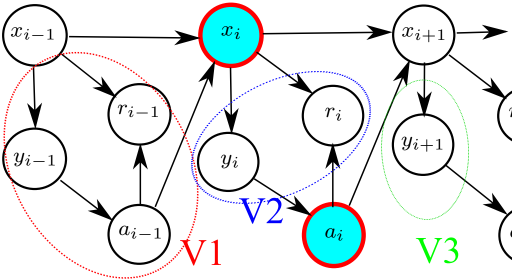


Tensor Moments

- $v_{i-1} \perp\!\!\!\perp v_i \perp\!\!\!\perp v_{i+1} | x_i, a_i$.
- $\mathbb{E}[v_1 \otimes v_2 \otimes v_3 | a_2 = l] = \sum_j \omega_\pi^{(l)} \cdot \mu_{1,j} \otimes \mu_{2,j} \otimes \mu_{3,j}$.
- Recover components of tensor decomposition.
- Simple manipulations to obtain parameters of POMDP.

Learning POMDPs Under Fixed Memoryless Policies

- Fixed memoryless policy π throughout learning process.



- $V_1^{(l)} = \mathbb{P}\big(\vec{y}_1, \vec{r}_1, a_1 | x_2, a_2 = l\big)$,
- $V_2^{(l)} = \mathbb{P}\big(\vec{y}_2, \vec{r}_2 | x_2, a_2 = l\big)$,
- $V_3^{(l)} = \mathbb{P}\big(\vec{y}_3 | x_2 = i, a_2 = l\big)$.

Outline

Learning POMDP model with spectral methods

Conditions for Learning POMDP

- Ergodic underlying Markov chain.
- Full column rank:

  Emission Matrix $O \in \mathbb{R}^{Y \times X}$
  Slices of Transition Tensor $P_a \in \mathbb{R}^{X \times X},\ a \in \mathcal{A}$

Learning POMDP model with spectral methods

Conditions for Learning POMDP

- Ergodic underlying Markov chain.
- Full column rank:

  Emission Matrix $O \in \mathbb{R}^{Y \times X}$
  Slices of Transition Tensor $P_a \in \mathbb{R}^{X \times X},\ a \in \mathcal{A}$

Sample Complexity

- Required: $T > \mathcal{O}(X^4) A \log(1/\delta)$,
- Relaxed stationarity condition, no need for mixing time

Learning Result Using Spectral Methods (Cont.)

- By probability at least $1 - 24A\delta$

$$\|\widehat{O}(:,i) - O(:,i)\|_1 = \mathcal{O}\left(\sqrt{\frac{Y\log(1/\delta)}{T_l}}\right),$$

$$\|\widehat{P}(\cdot,i,l) - P(\cdot,i,l)\|_1 = \mathcal{O}\left(\sqrt{\frac{Y\cdot X^2\log(1/\delta)}{T_l}}\right).$$

Learning + Planning in POMDPs

Tractable analysis by decoupling learning and planning.

Learning + Planning in POMDPs

Tractable analysis by decoupling learning and planning.

Episodic Learning

- Each episode, fixed policy π, collect samples.
- Learn Model Parameters.
- Update π.

Learning + Planning in POMDPs

Tractable analysis by decoupling learning and planning.

Episodic Learning

- Each episode, fixed policy π, collect samples.
- Learn Model Parameters.
- Update π.

UCRL: Upper Confidence Reinforcement Learning

- Episode length: Number of samples, doubling trick (at least samples for one action is doubled), ($\alpha = 2$)
- Update policy
  - All possible POMDPs.
  - Choose optimistic (stochastic) policy (oracle access assumed).

Learning Result Using Spectral Methods (Cont.)

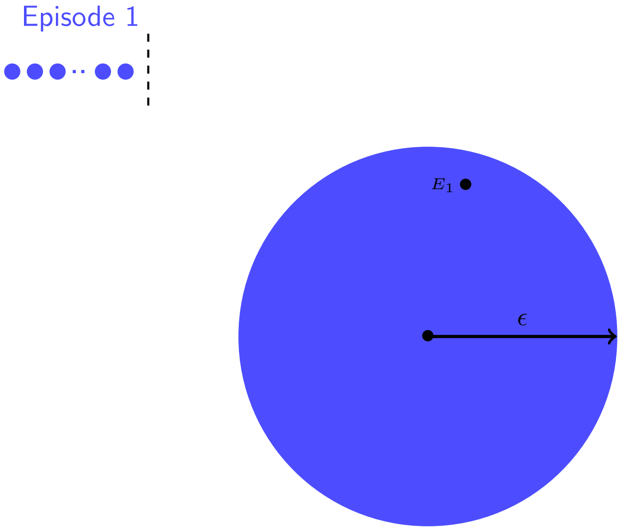
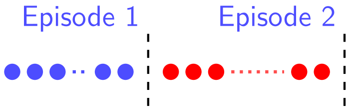


Figure from Hanie Sedghi's slides

Learning Result Using Spectral Methods (Cont.)



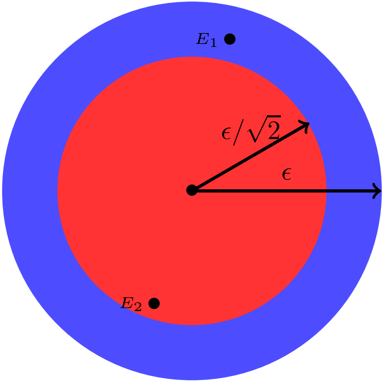


Figure from Hanie Sedghi's slides

Learning Result Using Spectral Methods (Cont.)

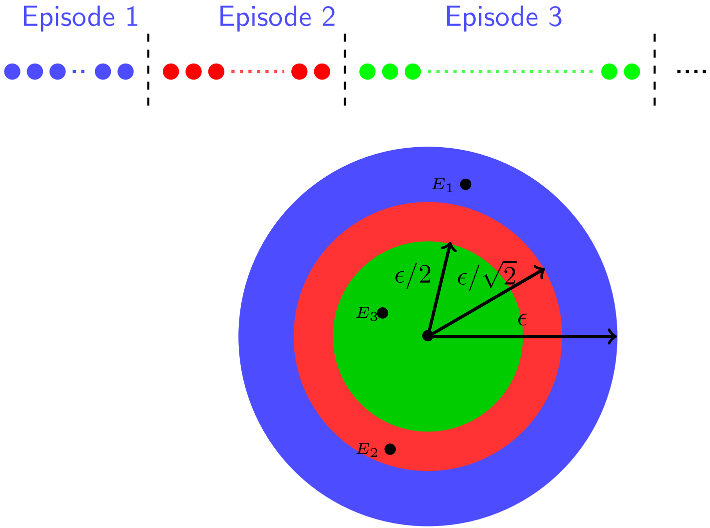


Figure from Hanie Sedghi's slides

Regret Bounds for POMDPs

- Cumulative regret: competing against best (stochastic) memoryless policy for the true model.

Regret Bounds for POMDPs

- Cumulative regret: competing against best (stochastic) memoryless policy for the true model.

$$\mathit{Reg}_T = T\ \eta^* - \sum_{t=1}^{T} r_t$$

- π: policy. $\mathcal{P}$: set of stochastic memoryless policies.
- D: Diameter of POMDP, τ: passing time

$$D := \max_{x,x'\in X, a,a'\in A} \min_{\pi\in\mathcal{P}} \mathbb{E}_\pi[\tau\left((x,a)\rightarrow(x',a')\right)]$$

Regret Bounds for POMDPs

Regret after T steps is:

$$\boxed{\text{Regret}(T) = \widetilde{\mathcal{O}}\left(D\sqrt{A \cdot Y \cdot X^3 \cdot T}\right)}$$

Regret Bounds for POMDPs

Regret after T steps is:

$$\boxed{\text{Regret}(T) = \widetilde{\mathcal{O}}\left(D\sqrt{A \cdot Y \cdot X^3 \cdot T}\right)}$$

- Compare to MDP ($Y = X$): $\text{Regret}(T) = \widetilde{\mathcal{O}}\left(\widetilde{D}\sqrt{A \cdot Y^2 \cdot T}\right)$.
- For MDP: diameter $\widetilde{D} := \max_{x,x' \in X} \min_{\pi} \mathbb{E}_{\pi}[\tau(x \to x')]$,
- Even better when $X^3 << Y$

Preliminary Experiments

Preliminary Experiments

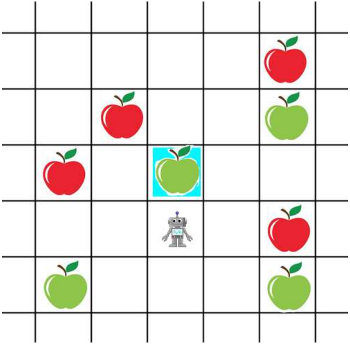
- Simple computer game

Preliminary Experiments

- Simple computer game

SM-UCRL-POMDP with (X=3) DQN with RMSprop $(10 \times 10 \times 10)$

Game Setting A = 4

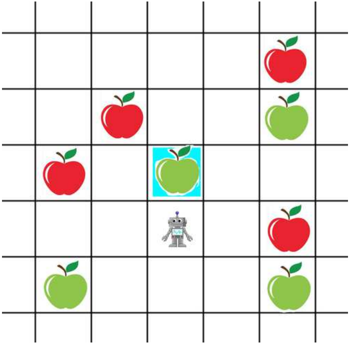


Preliminary Experiments

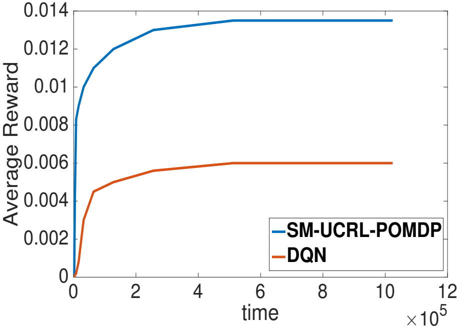
- Simple computer game

SM-UCRL-POMDP with (X=3) | DQN with RMSprop $(10 \times 10 \times 10)$

Game Setting A = 4



Performance

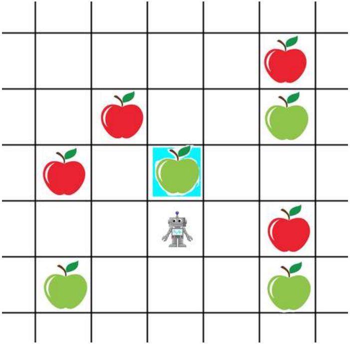


Preliminary Experiments

- Simple computer game

SM-UCRL-POMDP with (X=3)   DQN with RMSprop $(10 \times 10 \times 10)$

Game Setting A = 8

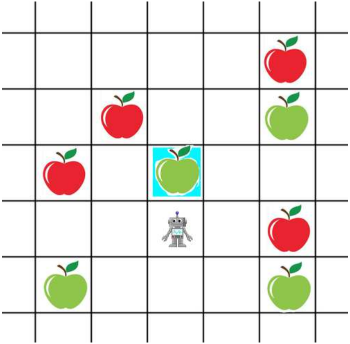


Preliminary Experiments

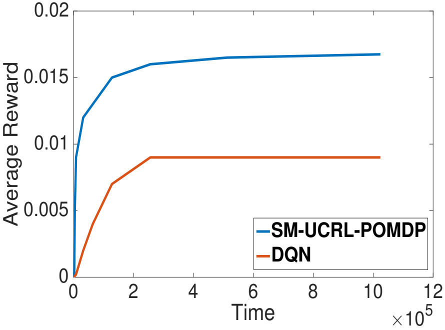
- Simple computer game

SM-UCRL-POMDP with (X=3) DQN with RMSprop ($10 \times 10 \times 10$)

Game Setting A = 8



Performance

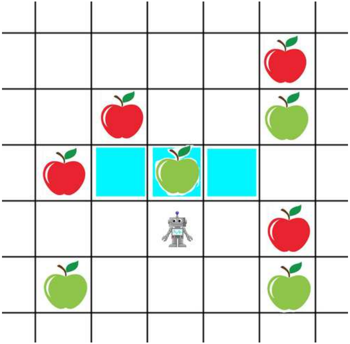


Preliminary Experiments

- Simple computer game

SM-UCRL-POMDP with (X=8)   DQN with RMSprop ($30 \times 30 \times 30$)

Game Setting, A = 8

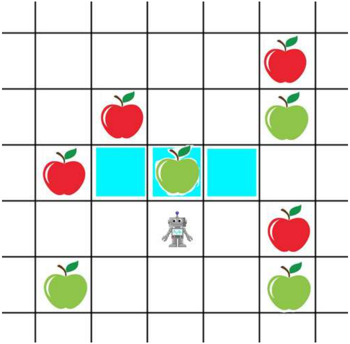


Preliminary Experiments

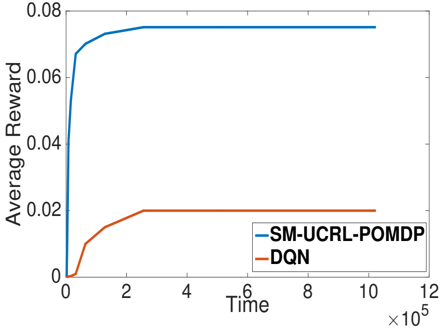
- Simple computer game

SM-UCRL-POMDP with (X=8) DQN with RMSprop ($30 \times 30 \times 30$)

Game Setting, $A = 8$



Performance



Outline

Moment Matrices and Tensors

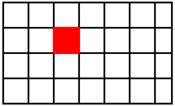
Multivariate Moments

- for random vectors y, y', y''

$$M_1 := \mathbb{E}[y], \quad M_2 := \mathbb{E}[y \otimes y'], \quad M_3 := \mathbb{E}[y \otimes y' \otimes y''].$$

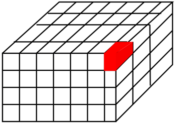
Matrix

- $\mathbb{E}[y \otimes y'] \in \mathbb{R}^{Y \times Y'}$ is a second order tensor.
- $\mathbb{E}[y \otimes y']_{i_1, i_2} = \mathbb{E}[y_{i_1} y'_{i_2}]$.
- For matrices: $\mathbb{E}[y \otimes y'] = \mathbb{E}[yy'^\top]$.



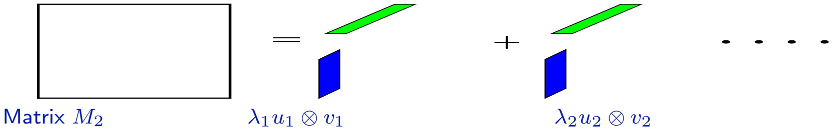
Tensor

- $\mathbb{E}[y \otimes y' \otimes y''] \in \mathbb{R}^{Y \times Y' \times Y''}$ is a third order tensor.
- $\mathbb{E}[y \otimes y' \otimes y'']_{i_1, i_2, i_3} = \mathbb{E}[y_{i_1} y'_{i_2} y''_{i_3}]$.



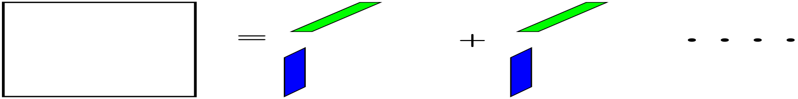
Spectral Decomposition of Matrices and Tensors

$$M_2 = \sum_i \lambda_i u_i \otimes v_i$$



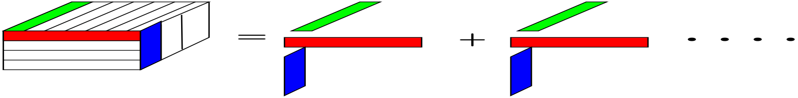
Spectral Decomposition of Matrices and Tensors

$$M_2 = \sum_i \lambda_i u_i \otimes v_i$$



Matrix M_2 $\lambda_1 u_1 \otimes v_1$ $\lambda_2 u_2 \otimes v_2$

$$M_3 = \sum_i \lambda_i u_i \otimes v_i \otimes w_i$$



Tensor M_3 $\lambda_1 u_1 \otimes v_1 \otimes w_1$ $\lambda_2 u_2 \otimes v_2 \otimes w_2$

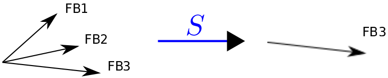
- $u \otimes v \otimes w$ is a rank-1 tensor since its $(i_1, i_2, i_3)^{\text{th}}$ entry is $u_{i_1} v_{i_2} w_{i_3}$.

Guaranteed Tensor Decomposition

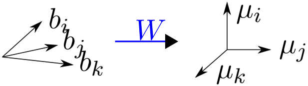
Non-orthogonal tensor

$$M_3 = \sum_i w_i [V_1]_i \otimes [V_2]_i \otimes [V_3]_i, \qquad M_2 = \sum_i w_i [V_1]_i \otimes [V_3]_i.$$

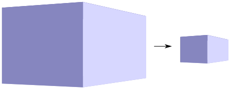
Symmetrizing



Whitening

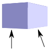


Dimension Reduction



Tensor M_3 Tensor M_3'

Tensor Power Method



Outline

Summary and Outlook

Summary

- Tensor methods: Novel Learning Method of POMDPs
- First methods to provide provable bounds for RL of POMDPs.
- UCRL of POMDPs.

Summary and Outlook

Summary

- Tensor methods: Novel Learning Method of POMDPs
- First methods to provide provable bounds for RL of POMDPs.
- UCRL of POMDPs.

Outlook

- Efficient deployment of tensor methods for RL. Comparison with deep neural network reinforcement learning in more complex environment
- Regret bound for limited memory policy, Belief based policy
- Optimal stochastic memoryless policy. *(Tomorrow at "Open Problem" session)*

Thank You!

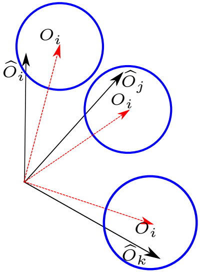
Learning Result Using Spectral Methods (Cont.)

- By probability at least $1 - 24A\delta$

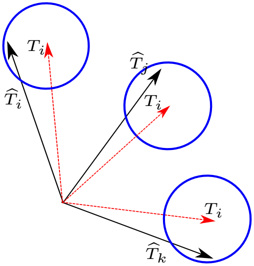
$$\|\widehat{O}(:,i) - O(:,i)\|_1 = \mathcal{O}\left(\sqrt{\frac{Y\log(1/\delta)}{T_l}}\right),$$

$$\|\widehat{T}(\cdot,i,l) - T(\cdot,i,l)\|_1 = \mathcal{O}\left(\sqrt{\frac{Y\cdot X^3\log(1/\delta)}{T_l}}\right).$$

Columns of O



Fibers of T



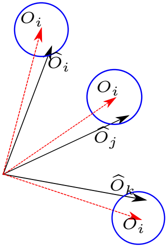
Learning Result Using Spectral Methods (Cont.)

- By probability at least $1 - 24A\delta$

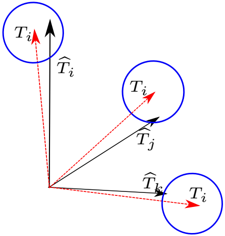
$$\|\widehat{O}(:,i) - O(:,i)\|_1 = \mathcal{O}\left(\sqrt{\frac{Y \log(1/\delta)}{T_l}}\right),$$

$$\|\widehat{T}(\cdot,i,l) - T(\cdot,i,l)\|_1 = \mathcal{O}\left(\sqrt{\frac{Y \cdot X^3 \log(1/\delta)}{T_l}}\right).$$

Columns of O



Fibers of T



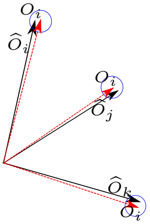
Learning Result Using Spectral Methods (Cont.)

- By probability at least $1 - 24A\delta$

$$\|\widehat{O}(:,i) - O(:,i)\|_1 = \mathcal{O}\left(\sqrt{\frac{Y \log(1/\delta)}{T_l}}\right),$$

$$\|\widehat{T}(\cdot,i,l) - T(\cdot,i,l)\|_1 = \mathcal{O}\left(\sqrt{\frac{Y \cdot X^3 \log(1/\delta)}{T_l}}\right).$$

Columns of O



Fibers of T