Online Expectation and Maximization for Model-Free Reinforcement Learning in POMDPs

Miao Liu

Joint work with Xuejun Liao and Lawrence Carin

April 25 2013

Main issues in sequential decision making under uncertainty

- The uncertainty resulting from imperfect sensors and actuators
- The tradeoff between Exploration and Exploitation
- Incompleteness of domain knowledge including state dynamics and observation models
- Scaling Issues caused by curses of dimensionality and history

Proposed solution

- POMDPs : partially observable Markov Decision processes
- Reinforcement Learning: learning from trial and error
- Online Learning: discarding the episode after the evaluation and memorizing the sufficient statistic

Markov Decision Processes (MDPs) $\langle S, A, T, R \rangle$

$\mathcal{S}$	a (finite) set of states
$\mathcal{A}$	a (finite) set of actions
$T : S \times A \to \Delta S$	a set of Markov transition functions
$T(s'\|s, a)$	the probability of transiting to state s' after taking action a in state s
$R : S \times A \to \mathbb{R}$	a reward function
$R(s, a)$	the reward obtained when action a is taken in state s.

- The Markov assumption: $T(s_t|s_{t-1}, s_{t-2}, \cdots, s_0, a) = T(s_t|s_{t-1}, a)$
- A policy $\pi : \mathcal{S} \to \mathcal{A}$
- The value function for an infinite-horizon MDP:

$$V^{\pi}(s_0) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(\pi(s_t), s_{t+1})\right]. \tag{1}$$

- Optimal policy defined by the Bellman equation (Bellman 1957)

$$\pi(s) = \arg\max_a V(s) \tag{2}$$

$$V(s) = R(s, a) + \gamma \sum_{s'} T(s'|s, a) V(s') \tag{3}$$

- Given the MDP, the optimal policy can be solved by dynamic programming.

Partially Observable Markov Decision Processes (POMDPs) $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, T, \Omega, R \rangle$

S, A, T, R	the same as for MDP
$\mathcal{O}$	a finite set of observations
$\Omega : S \times A \to O$	a set of observation functions
$\Omega(o\|a, s')$	the probability of observing o after taking action a and transiting to state s'

- o provides partial information about the underlying state s
- POMDP = HMM + control
- A belief state is a probability distribution over states that can summarize the knowledge of the agent at a given point.

$$b(s_t) = Pr(s_t = s|s_0, a_1, o_1, a_2, o_2, \cdots, a_{t-1}, o_{t-1}) \tag{4}$$

- Bayesian updating of beliefs

$$b(s') = Pr(s'|b, a, o) = \frac{\sum_{s \in \mathcal{S}} b(s) T(s'|s, a) \Omega(o|a, s')}{p(o|b, a)} \tag{5}$$

Value Function and hardness for solving POMDPs

- A POMDP can be formulated as a belief-state MDP with Value function (infinite horizon)

$$V^{\pi}(b) = R(b,a) + \gamma \sum_{a \in \mathcal{A}} p^{\pi}(a|b) \sum_{o \in \mathcal{O}} p(o|b,a) V^{\pi}(b') \tag{6}$$

  where $R(b,a) = \sum_{s \in S} b(s) R(s,a)$ is the expected reward and

$$T(b'|b,a) = \sum_{o \in \mathcal{O}} p(b'|b,a,o) P(o|b,a) = \begin{cases} p(o|b,a) & \text{, if } b' = Pr(s'|b,a,o) \\ 0 & \text{, otherwise} \end{cases}$$

  is the belief state transition function.

- Dynamic programming can be used to find a solution to the belief state MDP
- The optimal value function of a (finite-horizon) POMDP is piecewise linear and convex (PWLC) [Smallwood and Sondik 1973], hence linear programming can be applied to solve the optimal policy (with PSPACE-comxplexity, if the horizon is smaller than $|\mathcal{S}|$)
- For infinie horizon POMDPs,
  - The hardness is undecidable
  - Point-based methods are the most efficient (approximate solution)
- Solving planning problems need the access to the POMDP model, which is not always available

Reinforcement learning strategies in POMDPs

- Model-based: Learn the POMDP model $\mathcal{M}$ first, then solve the control policy based on $\mathcal{M}$
- Model-free: the agent learns the policy directly from the experience, skipping the model-learning step.

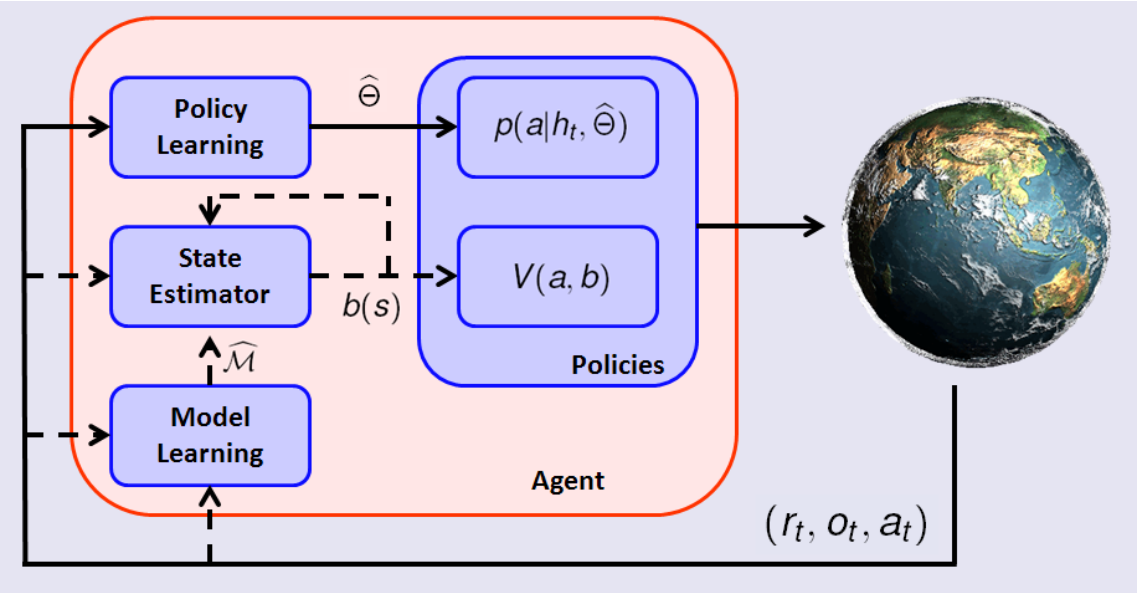


Figure: The planning and learning framework for POMDPs

Regionalized Policy Representation (RPR) $(\mathcal{A}, \mathcal{O}, \mathcal{Z}, W, \mu, \pi)$ [Li et al., JMLR09]

$\mathcal{A}$	a finite set of actions
$\mathcal{O}$	a finite set of observations
$\mathcal{Z}$	a finite set of decision states
W	a set of Markov transition matrices
$W_{z'}^{zao}$	the probability of transiting from z to z' when taking action a in z results in observation o
μ	the probability of initially being in z
π	a set of stochastic local policies
π_a^z	the probability of taking action a in z

- The RPR policy

$$p(a_t|h_t,\Theta)=p(a_t|a_{0:t-1},o_{1:t},\Theta)=\frac{p(a_{0:t}|o_{1:t},\Theta)}{p(a_{0:t-1}|o_{1:t-1},\Theta)} \tag{7}$$

- $p(a_{0:t}|o_{1:t},\Theta)$ results from

$$p(a_{0:t},z_{0:t}|o_{1:t},\Theta)=\mu_{z_0}\pi_{a_0}^{z_0}\prod_{\tau=1}^{t}W_{z_\tau}^{z_{\tau-1}a_{\tau-1}o_\tau}\pi_{a_\tau}^{z_\tau}, \tag{8}$$

  by marginalizing out latent decision states $z_{0:t}$.
- $\Theta=\{\pi,\mu,W\}$ is learned empirically from experiences, instead of being computed from an underlying POMDP model.

Empirical Value Function and its lower bound

- Given episodes $\mathcal{D}^{(K)} = \{(a_0^k r_0^k o_1^k a_1^k r_1^k \cdots o_{T_k}^k a_{T_k}^k r_{T_k}^k)\}_{k=1}^K$, define the empirical value function

$$\widehat{V}(\mathcal{D}^{(K)}; \Theta) \stackrel{def.}{=} \frac{1}{K} \sum_{k=1}^{K} \sum_{t=0}^{T_k} \gamma^t r_t^k \frac{\prod_{\tau=0}^{t} p(a_\tau^k | h_\tau^k, \Theta)}{\prod_{\tau=0}^{t} p^{\Pi}(a_\tau^k | h_\tau^k)} \tag{9}$$

- $\lim_{K\to\infty} \widehat{V}(\mathcal{D}^{(K)}; \Theta) = \mathbb{E}_{episodes|\mathcal{M},\Theta}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$.
- The lower bound of log empirical value function

$$\ln \widehat{V}(\mathcal{D}^{(K)}; \Theta) \geq \sum_{k,t,z_{0:t}^k} \frac{q_t^k(z_{0:t}^k|\widetilde{\Theta})}{K} \ln \frac{\tilde{r}_t^k p(a_{0:t}^k, z_{0:t}^k | o_{1:t}^k, \Theta)}{q_t^k(z_{0:t}^k|\widetilde{\Theta})} \tag{10}$$

$$\stackrel{def.}{=} \mathrm{lb}(\Theta|\widetilde{\Theta}) \tag{11}$$

  where $\tilde{\gamma}_t^k = \frac{\gamma^t r_t^k}{\prod_{\tau=0}^{t} p^{\Pi}(a_\tau^k | h_\tau^k)}$

- Define $\mathcal{F} = \Big\{\Theta = (\mu, \pi, W) : \sum_{u=1}^{|\mathcal{Z}|} \mu_u = 1, \sum_{a=1}^{|\mathcal{A}|} \pi_u^a = 1, \sum_{u=1}^{|\mathcal{Z}|} W_{vu}^{ao} = 1, v = 1 \cdots |\mathcal{Z}|, a = 1 \cdots |\mathcal{A}|, o = 1 \cdots |\mathcal{O}|\Big\}$
- $\Theta^{(m+1)} = \arg\max_{\Theta\in\mathcal{F}} \mathrm{lb}(\Theta|\Theta^{(m)})$
- $\{\Theta^{(m)}\}_{m\geq 0}$ monotonically increases (11), until convergence to a maxima

Batch EM algorithm [Liao et al 07, Li et al 09 JMLR]

- Update decision states (E-step)

$$q_t^k(z_{0:t}^k) \overset{def.}{=} \frac{\tilde{r}_t^k p(a_{0:t}^k, z_{0:t}^k | o_{1:t}^k, \widetilde{\Theta})}{\hat{V}(\mathcal{D}^{(K)}; \widetilde{\Theta})} = \nu_t^k p(z_{0:t}^k | a_{0:t}^k, o_{1:t}^k, \widetilde{\Theta}), \tag{12}$$

where ν_t^k is the recomputed reward with

$$\nu_t^k = \frac{\tilde{\gamma}_t^k p(a_{0:t}^k | o_{1:t}^k, \widetilde{\Theta})}{\hat{V}(\mathcal{D}^{(K)}; \widetilde{\Theta})}, \forall\, t, k, \tag{13}$$

and $p(z_{0:t}^k | a_{0:t}^k, o_{1:t}^k, \widetilde{\Theta})$ is computed by using

$$\xi_{t,\tau}^k(i,j) = p(z_\tau^k = i, z_{\tau+1}^k = j | a_{0:t}^k, o_{1:t}^k, \widetilde{\Theta}), \tag{14}$$

$$\phi_{t,\tau}^k(i) = p(z_\tau^k = i | a_{0:t}^k, o_{1:t}^k, \widetilde{\Theta}), \tag{15}$$

- Update policy parameters (M-step)

$$W(u,a,o,v) = \frac{\sum_{k,t,\tau} \nu_t^k \xi_{t,\tau}^k(u,v) \delta(a_{\tau,a}^k) \delta(o_{\tau+1}^k, o)}{\sum_{u,k,t,\tau} \nu_t^k \xi_{t,\tau}^k(u,v) \delta(a_\tau^k, a) \delta(o_{\tau+1}^k, o)}$$

$$\pi(u,a) = \frac{\sum_{k,t,\tau} \nu_t^k \phi_{t,\tau}^k(u) \delta(a_\tau^k, a)}{\sum_{a,k,t,\tau} \nu_t^k \phi_{t,\tau}^k(u) \delta(a_\tau^k, a)}$$

$$\mu(u) = \frac{\sum_{k,t} \nu_t^k \phi_{t,0}^k(u)}{\sum_{u,k,t} \nu_t^k \nu_{t,0}^k(u)} \tag{16}$$

Online Expectation Maximization Algorithm [Liu et al. IJCAI13]

- Let $\boldsymbol{\theta} = [\boldsymbol{\mu}, \boldsymbol{\pi}, \boldsymbol{w}]^T$ and $\boldsymbol{s} = [\upsilon, \boldsymbol{\rho}, \boldsymbol{\omega}]^T$ be vectorized RPR parameters and sufficient statistics respectively
- Define $\psi(\boldsymbol{\theta}) = \ln \boldsymbol{\theta} - \boldsymbol{C}\boldsymbol{\theta}$, with $\boldsymbol{C}$ a all one block diagonal matrix encoding the normalization constraint
- The objective (lower bound): $f(\hat{\boldsymbol{\theta}}; \boldsymbol{s}) = \boldsymbol{s}^T \psi(\hat{\boldsymbol{\theta}})$
- E-step: $\hat{\boldsymbol{s}}(\mathcal{D}^{(n)}, \boldsymbol{\theta}_{n-1}) = \sum_{k=1}^{n} \alpha_k \prod_{j=k+1}^{n} (1-\alpha_j) \boldsymbol{s}(\varepsilon_k, \boldsymbol{\theta}_{k-1})$, with the recursive definition

$$\hat{\boldsymbol{s}}_k = (1-\alpha_k)\hat{\boldsymbol{s}}_{k-1} + \alpha_k \boldsymbol{s}(\varepsilon_k, \boldsymbol{\theta}_{k-1}). \tag{17}$$

- M-step: $\boldsymbol{\theta}_k = \arg\max f(\hat{\boldsymbol{\theta}}; \hat{\boldsymbol{s}}_k)$, which is equivalent to solving

$$\nabla_{\boldsymbol{\theta}} \psi^T(\boldsymbol{\theta}) \hat{\boldsymbol{s}}_k = \boldsymbol{0}. \tag{18}$$

- If $\sum_{n=1}^{\infty} \alpha_n = \infty$ and $\sum_{n=1}^{\infty} \alpha_n^2 < \infty$, then $\lim_{n\to\infty} \mathrm{dist}(\hat{\boldsymbol{s}}_n, \Delta) = 0$, where Δ is the optimal batch model policy

The dual-policy RPR for Online Learning

- Behavior Policy: $p^{\Pi}(a|h) = p(y=0|h)p(a|h,\Theta) + p(y=1|h)/|\mathcal{A}|$.
- $p(a|h,\Theta)$ controls regular action selection.
- $p(y|h) = \sum_{z\in\mathcal{Z}} \sigma_y^z p(z|h), \forall\, h$ controls exploitation ($y=0$) and exploration ($y=1$) [Cai et al., NIPS09]
  - $\sigma_0^z = p(y=0|z)$

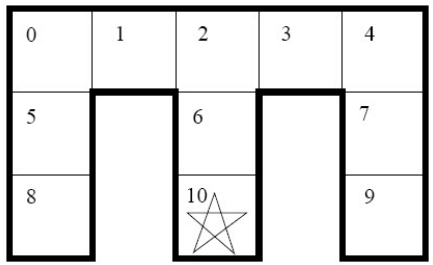
$$\sigma_0^z \sim \text{Beta}(u_0^z, u_1), \text{ with } \sigma_1^z = 1-\sigma_0^z, \ \forall\, z \in \mathcal{Z}, \tag{19}$$

where $u_1 > 1$ is a given constant and $\{u_0^z\}_{z=1}^{|\mathcal{Z}|}$ are updated using the rule,

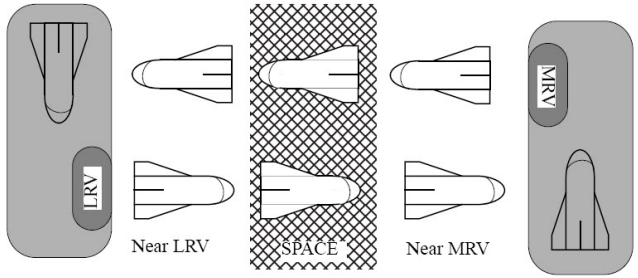
$$u_0^i = \sum_{k=1}^{K} \sum_{t=0}^{T_k} \hat{\nu}_t^k \sum_{\tau=0}^{t} \phi_{t,\tau}^k(i), \ \forall\, i \in \mathcal{Z}, \tag{20}$$

  - u_1 defines the total reward required in z for the agent to stop exploration in z.
  - With a sufficiently large u_1, the RPR is guaranteed to converge to the optimal policy (assuming $|\mathcal{Z}|$ is appropriate).

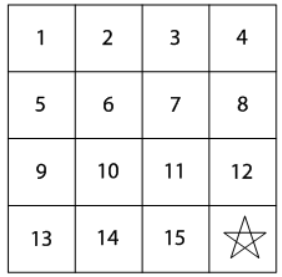
Benchmark problems



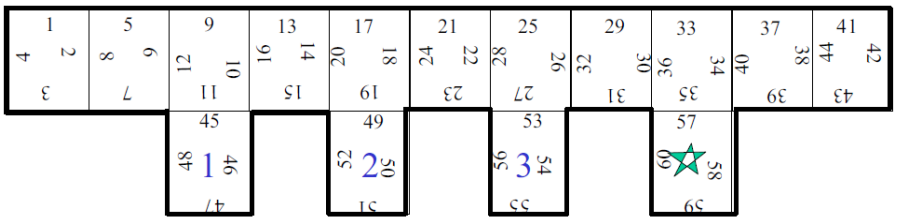
(a) Cheese maze



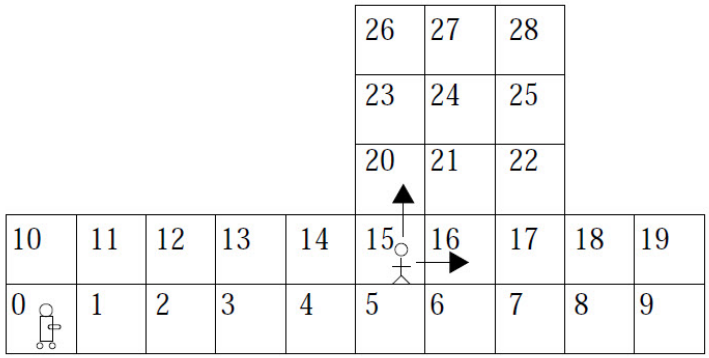
(b) Shuttle docking



(c) 4×4 grid



(d) Hallway



(e) Tag

Table: Five benchmark POMDP problems

Name	Shuttle	Cheese maze	4×4 grid	Hallway	Tag
$\lvert\mathcal{A}\rvert$	8	4	4	5	5
$\lvert\mathcal{O}\rvert$	3	7	2	21	30
Number of States	3	11	16	60	870

Online RPR versus Batch RPR

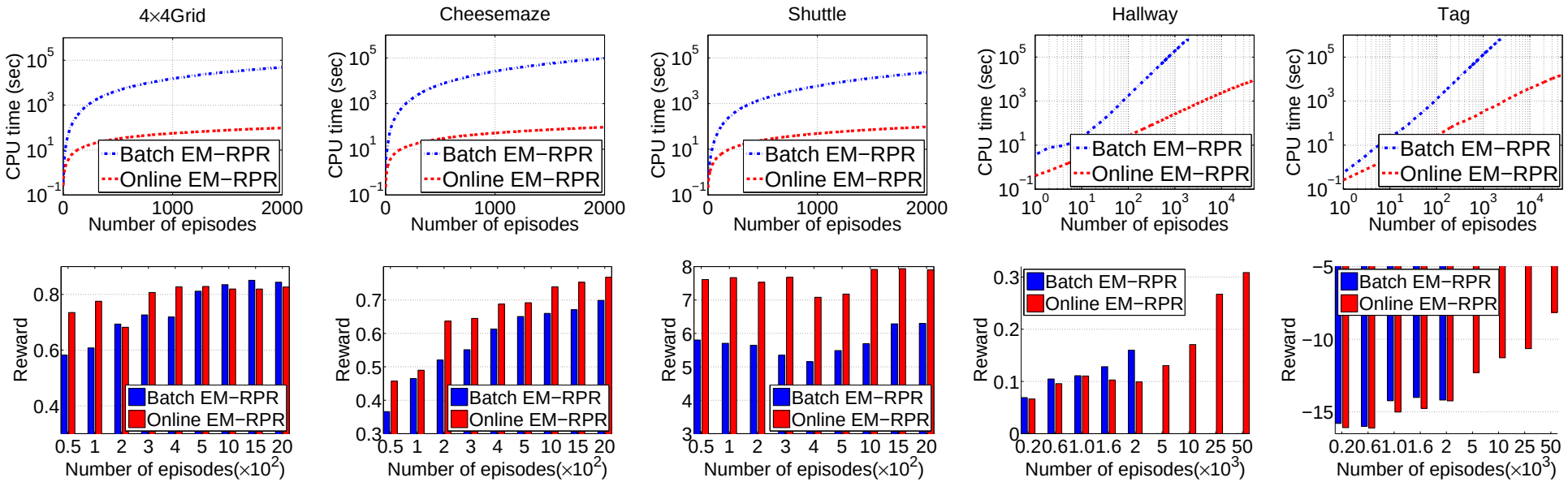


Table: Time complexity

	Densely rewarded	Terminally rewarded	Approximate
Batch	$O(\|\mathcal{A}\|\|\mathcal{O}\|\|\mathcal{Z}\|^2 T^2 n^2)$	$O(\|\mathcal{A}\|\|\mathcal{O}\|\|\mathcal{Z}\|^2 T n^2)$	$O(n^2)$
Online	$O(\|\mathcal{A}\|\|\mathcal{O}\|\|\mathcal{Z}\|^2 T^2 n)$	$O(\|\mathcal{A}\|\|\mathcal{O}\|\|\mathcal{Z}\|^2 T n)$	$O(n)$

Comparison with Other Methods

RPR vs U-tree(McCallum, PhD Thesis 95, Zheng and Cho, Neural Process Lett 11)

Table: Averaged reward earned by the competing algorithms on small problems. The parenthesized integers indicate the numbers of nodes used in the U-tree algorithms

Methods	Shuttle	Cheese Maze	4×4 Grid
U-Tree	**1.833 (62)**	0.184 (53)	0.179 (47)
Modified U-Tree	1.820 (25)	**0.184 (19)**	**0.179 (20)**
RPR ($\lvert\mathcal{Z}\rvert = \mathbf{10}$)	1.780$\pm$ 0.040	0.1556$\pm$ 0.049	**0.222$\pm$0.020**
RPR ($\lvert\mathcal{Z}\rvert = \mathbf{20}$)	1.786 $\pm$0.068	0.165$\pm$0.024	**0.226$\pm$0.020**
RPR ($\lvert\mathcal{Z}\rvert = 30$)	1.780$\pm$0.069	0.154$\pm$0.051	0.214$\pm$0.052

- All these methods are model-free
- They learn the policy without using POMDP model
- RPR uses smaller number of decision states and achieves comparable performance

RPR vs iPOMDP (Doshi-Velez NIPS09) and Policy Prior (PP, Doshi-Velez et al. NIPS10)

Table: Cumulative Reward

Name	RPR (10)	RPR (40)	iPOMDP	PP(1)	PP(2)	PP(3)
Hallway	6.0±2.8	3.0±1.3	0.2	1.4	1.6	6.6
Tag	-4908±95	-4987±13	-16000	-6900	-7400	-3500

- iPOMDP and PP are model-based methods
- iPOMDP begins with random actions for collecting samples
- PP can use the expert knowledge (data) for solving the policy
- After the same learning steps, RPR achieves better performance than iPOMDP and is compatible to PP

RPR vs MDP-EM (Vlassis and Toussaint, ICML09)

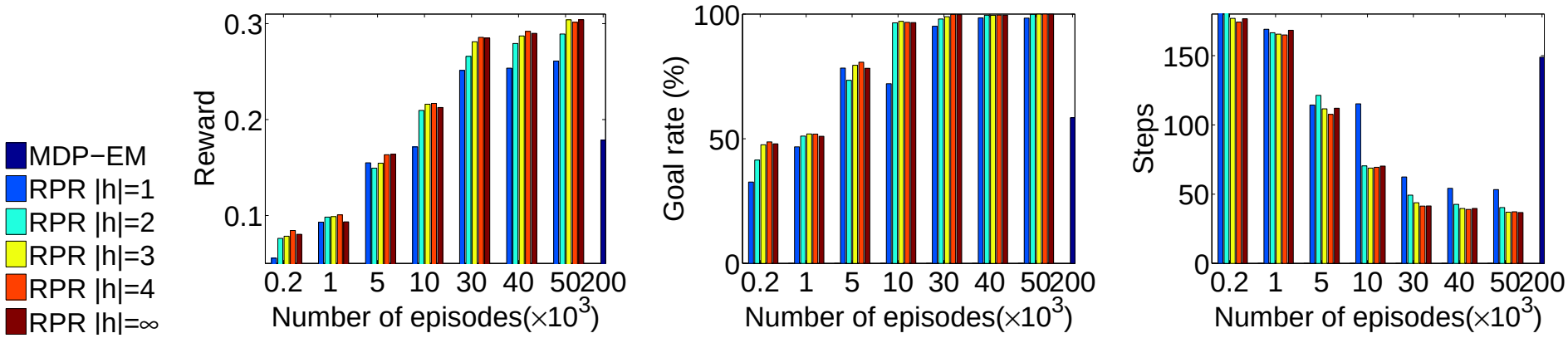


Figure: The results on Hallway. Comparison between MDP-EM and Online EM-RPR. $|h|$ is the length of history (number of consecutive observations) used by online EM-RPR for decision making.

- Both are model-free and use EM algorithm, but with different learning objective function
- MDP-EM is developed based on MDP formulation, which treat observation as state directly
- RPR is directly developed for POMDPs and the policy condition on all history (when $|h| = 1$ the policy is the same as MDP-EM)

The Performance of RPR as a Function of $|Z|$

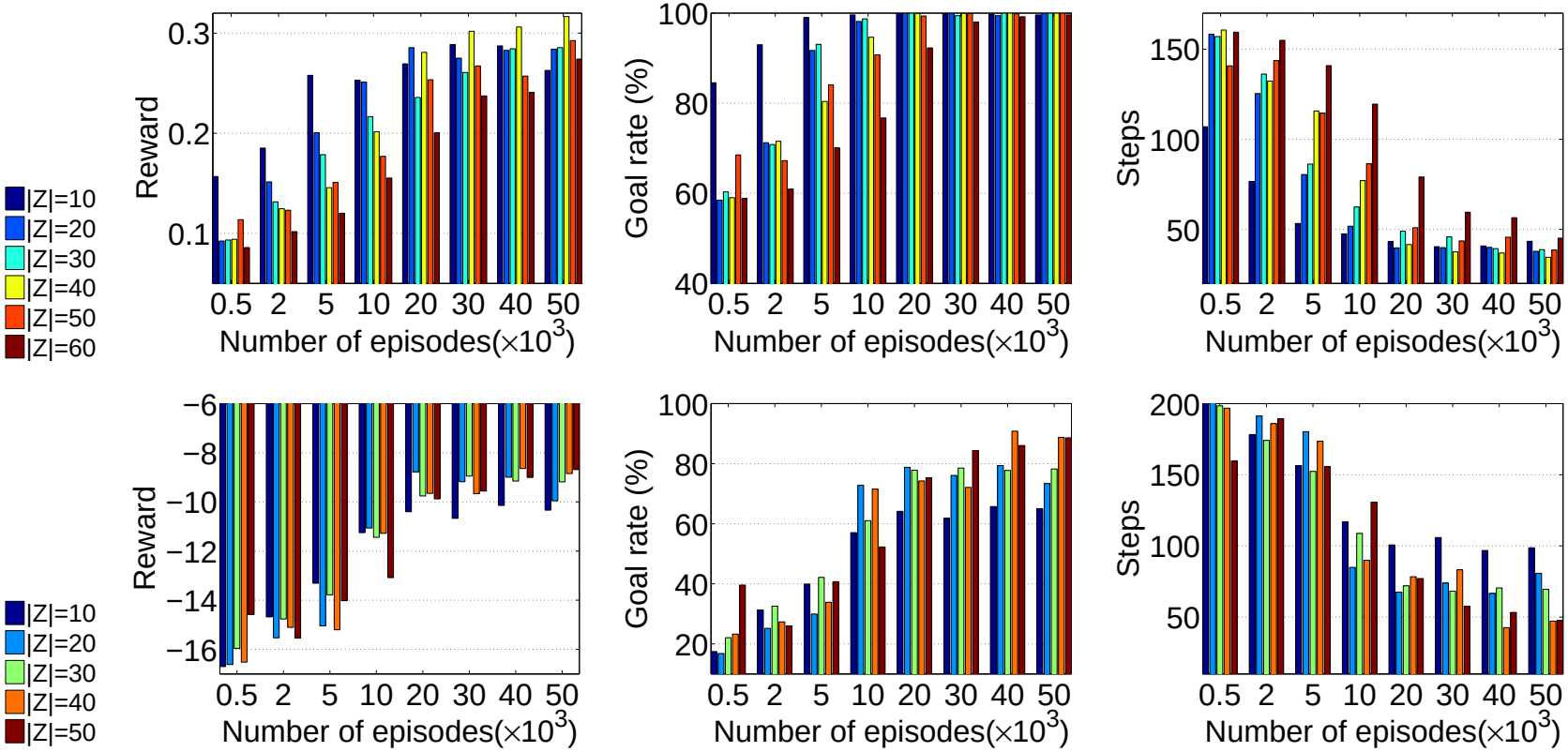


Figure: The results on Hallway (top) and Tag (bottom) produced by the RPR learned with the online EM algorithm.

- The performance for the online-learned policy as a function of $|Z$ is robust over a wide range of $|Z|$.
- When $|Z|$ is too small or too large, the performance is poor
- Automatic inference of $|Z|$ can be performed using the infinite- RPR framework in [Liu et al., 2011].

Summary

online EM algorithm for RPR policy Learning

- Updates control policy on the fly
- Converges to the batch model solution
- A competent candidate for solving large scale RL problems (with partially observability).

On going and future works

- Online inference for $|\mathcal{Z}|$
  - online learning with Bayesian nonparametric methods
- Multi-agent reinforcement learning
  - Decentralized POMDPs
  - Convergence analysis
- Test on real problems

References

1. R. A. McCallum. Reinforcement Learning with Selective Attention and Hidden State. PhD thesis, Department of Computer Science, University of Rochester, 1995.
2. H. Li, X. Liao, and L. Carin. Multi-task reinforcement learning in partially observable stochastic environments. JMLR, 10:1131-1186, 2009.
3. C. Cai, X. Liao, and L. Carin. Learning to explore and exploit in POMDPs. NIPS, pages 198-206, 2009.
4. F. Doshi-Velez, D. Wingate, N. Roy, and J. Tenenbaum. The infinite partially observable markov decision process. NIPS, pages 477-485, 2009.
5. N. Vlassis and M. Toussaint. Model-free reinforcement learning as mixture learning. ICML, pages 1081-1088, 2009.
6. F. Doshi-Velez. Nonparametric bayesian policy priors for reinforcement learning. NIPS, pages 532-540, 2010.
7. L. Zheng and S. Cho. A modified memory-based reinforcement learning method for solving pomdp problems. Neural Process Lett, (33):187-200, 2011.
8. M. Liu, X. Liao, and L. Carin. The infinite regionalized policy representation. ICML, pages 769-776, 2011.
9. M. Liu, X. Liao, and L. Carin. Online Expectation Maximization for Reinforcement Learning in POMDPs (To appear in IJCAI 2013)