

# **Reinforcement Learning & Life-Long Learning for POMDPs**

*Lawrence Carin, Hui Li and Xuejun Liao*

Electrical & Computer Engineering  
Duke University  
[www.ee.duke.edu/~lcarin](http://www.ee.duke.edu/~lcarin)

# Outline

- Review of model-based POMDPs and policy design, motivation for RL
- Idea behind regionalized policy representation (RPR) for RL in POMDPs
- RPR implementation
- Multi-task and life-long learning with POMDPs
- Example results
- Future directions

## Belief State as a Sufficient Statistic

- The belief state quantifies the probability that the sensor is in state  $s$  given a sequence of  $T$  actions and corresponding observations
- The belief state at time  $T$  is a sufficient statistic for all actions and observations up to that point

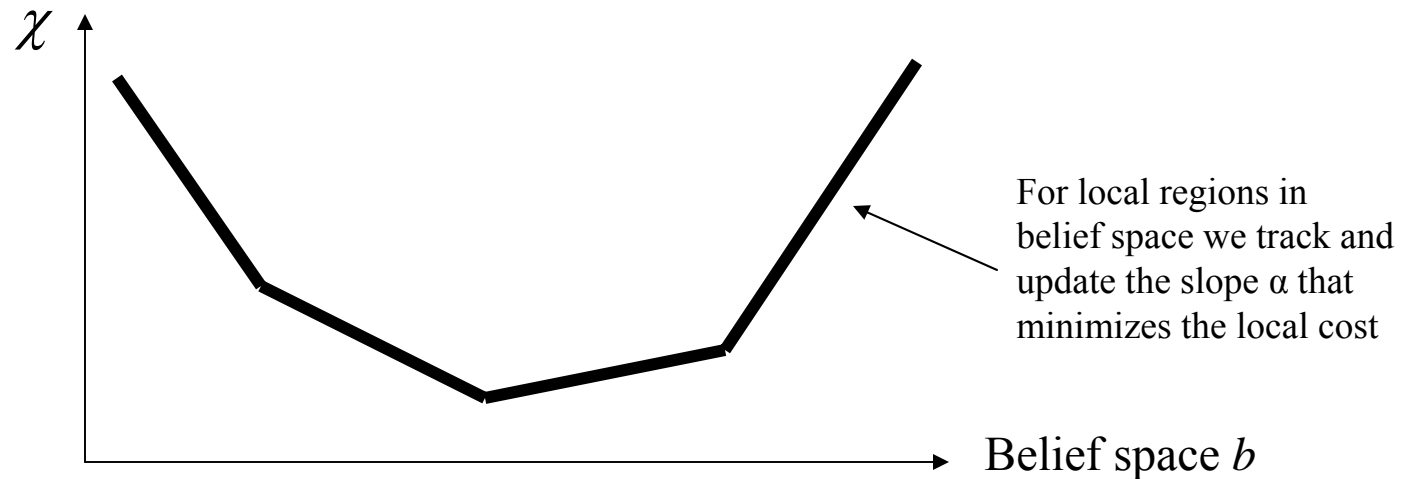
$$b_T(s|o_1, \dots, o_T, a_1, \dots, a_T) = \Pr(s|o_T, a_T, b_{T-1})$$

- Very important for practical implementation: Needn't store all previous actions & observations
- Belief state computed readily, using underlying target POMDP model

$$\begin{aligned} b_T(s') &= \frac{\Pr(o_T|s', a_T, b_{T-1}) \Pr(s'|a_T, b_{T-1})}{\Pr(o_T|a_T, b_{T-1})} \\ &= \frac{\Pr(o_T|s', a_T, b_{T-1}) \sum_s \Pr(s'|a_T, b_{T-1}, s) \Pr(s|a_T, b_{T-1})}{\Pr(o_T|a_T, b_{T-1})} \\ &= \frac{p(o_T|s', a_T) \sum_s p(s'|a_T, s) b_{T-1}(s)}{\Pr(o_T|a_T, b_{T-1})} \end{aligned}$$

# Value Function & Value Iteration

- The cost function is linear in the belief state, which implies that the cost function is a piecewise linear concave problem in the belief-space simplex



$$\chi_t(b) = \min_{\alpha \in C_t} \sum_{s \in S} \alpha(s) b(s)$$

$$\chi_t(b) = \min_{a \in A} \left[ C(b, a) + \gamma \sum_{o \in O} \min_{\alpha \in C_{t-1}} \sum_{s \in S} \sum_{s' \in S} p(s'|s, a) p(o|s', a) \alpha(s') b(s) \right]$$

- Value iteration becomes problem of learning the belief-state local slopes  $\alpha(b)$ , for each of which there is an optimal action (policy) – Policy learned by tracking slopes approximately

# Model-Based Policy

- If it is assumed that the underlying POMDP model is known, then one may compute the belief state  $\mathbf{b}$
- In many practical sensing applications the model is unknown
- One could in principle first obtain some “training” data and design a model
- This model may then be employed for model design
- Using this model belief states may be computed, and the policy is a mapping from belief states to actions
- Can we learn the policy directly based on experiences: reinforcement learning

# Outline

- Review of model-based POMDPs and policy design, motivation for RL
- Idea behind regionalized policy representation (RPR) for RL in POMDPs
- RPR implementation
- Multi-task and life-long learning with POMDPs
- Example results
- Future directions

# RL and Stochastic POMDP Policies - 1/2

- When you have a model, you can calculate the belief state deterministically, and the policy is a deterministic mapping from belief states to actions
- When developing model-free policies, we don't have the underlying POMDP model, we just have previous experiences (sequences of actions, observations and rewards)
- We may now consider regions in belief-state space
- Based upon a given history (sequence of actions, observations and rewards) we think abstractly of being within a *region* in belief space (not at a point)
- Each such region may be thought of as a “machine” state – *not* a true world state

## RL and Stochastic POMDP Policies - 2/2

- Policy mapping from belief states to actions is deterministic
- Since for machine states we do not explicitly know what the belief state is, when in a machine state we have a *stochastic* mapping to actions
- In our RL algorithm we have a probability of being in a particular machine state based on a previous history
- As we make series of actions and observations, walk between machine states modeled as a Markov process
- Do not explicitly require that the actual world is a POMDP, only that it is a partially observable stochastic system

# Outline

- Review of model-based POMDPs and policy design, motivation for RL
- Idea behind regionalized policy representation (RPR) for RL in POMDPs
- RPR implementation
- Multi-task and life-long learning with POMDPs
- Example results
- Future directions

# Regionalized Policy Representations (RPR)

- Introduction to reinforcement learning in POMDPs
- Idea of the regionalized policy representation (RPR)
- Algorithm of the RPR
- Experimental results of the RPR
- Conclusions on the RPR

# Regionalized Policy Representations (RPR)(Cont'd)

Difficulties in model-free reinforcement learning in POMDPs

- Belief state is not available (**not observable**)
- Carrying a long history of actions and observations is cumbersome and inefficient
- Finding a compact way to represent the history is the key thing to the model-free reinforcement learning in POMDPs

# Regionalized Policy Representations (RPR)(Cont'd)

Idea of the regionalized policy representation (RPR)

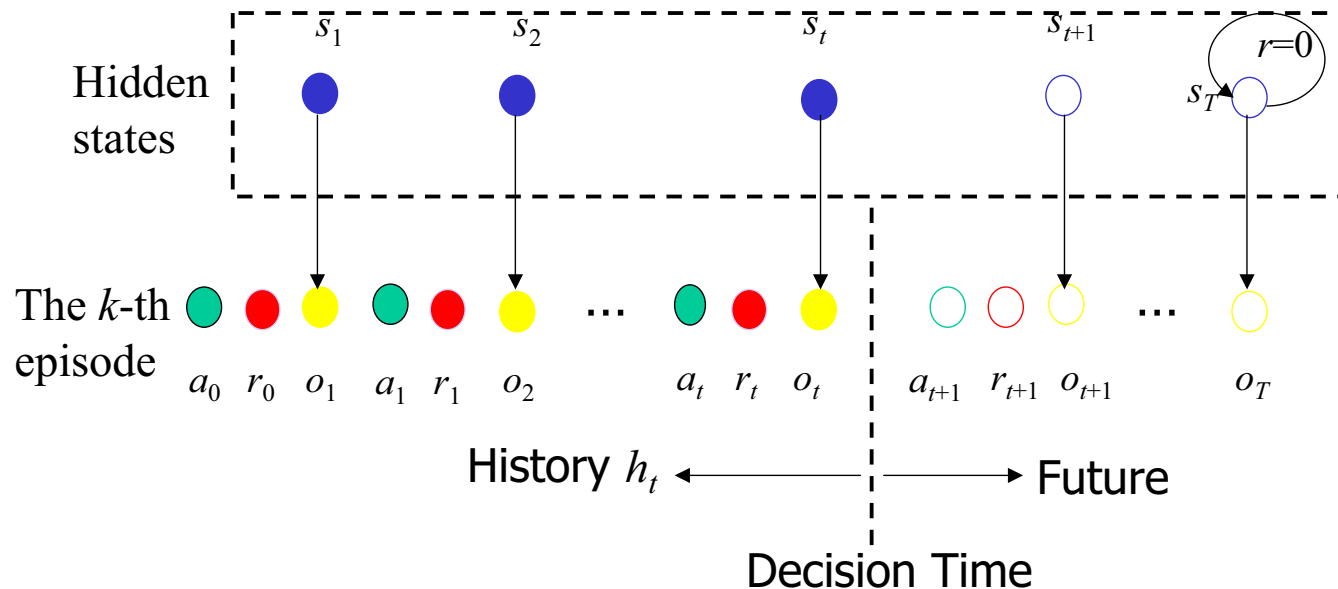
- The goal is to find a policy  $p(a|h) : h \rightarrow a$  so as to maximize the expected discounted future reward  $\mathbb{E}_p(\sum_t \gamma^t r_t)$
- We parameterize the policy  $p(a|h)$  with  $p(a|h, \Theta)$
- The empirical value function is represented by

$$\widehat{V}(\mathcal{D}^{(K)}; \Theta) \stackrel{def.}{=} \frac{1}{K} \sum_{k=1}^K \sum_{t=0}^{T_k} \frac{\gamma^t r_t^k}{\prod_{\tau=0}^t p^{\Pi}(a_{\tau}^k | h_{\tau}^k)} \prod_{\tau=0}^t p(a_{\tau}^k | h_{\tau}^k, \Theta) \quad (17)$$

# Regionalized Policy Representations (RPR)(Cont'd)

Empirical value function  $\widehat{V}(\mathcal{D}^{(K)}; \Theta)$

- Episodes:  $\mathcal{D}^{(K)} = \{(a_0^k r_0^k o_1^k a_1^k r_1^k \cdots o_{T_k}^k a_{T_k}^k r_{T_k}^k)\}_{k=1}^K$
- $\Pi$ : Episodes are obtained by following policy  $\Pi$ ; the learning of the policy is **off-policy** learning



# Regionalized Policy Representations (RPR)(Cont'd)

- $\hat{V}(\mathcal{D}^{(K)}; \Theta)$  implements  $\mathbb{E}_p(\sum_t \gamma^t r_t)$  using Monte Carlo integration with importance sampling, which guarantees to converge almost surely to the true value function  $\mathbb{E}_p(\sum_t \gamma^t r_t)$

$$\begin{aligned}\mathbb{E}_{p(x)}(f(x)) &= \int_x f(x)p(x)dx \\ &\approx \frac{1}{K} \sum_{k=1}^K f(x_k), x_k \text{ is drawn from } p(x)\end{aligned}\quad (18)$$

$$\begin{aligned}\mathbb{E}_{p(x)}(f(x)) &= \int_x \frac{f(x)p(x)}{g(x)}g(x)dx \\ &\approx \frac{1}{K} \sum_{k=1}^K \frac{f(x_k)p(x_k)}{g(x_k)}, x_k \text{ is drawn from } g(x)\end{aligned}\quad (19)$$

# Regionalized Policy Representations (RPR)(Cont'd)

Parametric form of  $p(a|h, \Theta)$

$$p(a_\tau|h_\tau, \Theta) = \frac{p(a_{0:\tau}|o_{1:\tau}, \Theta)}{p(a_{0:\tau-1}|o_{1:\tau-1}, \Theta)} \quad (20)$$

$$\begin{aligned} \prod_{\tau=0}^t p(a_\tau|h_\tau, \Theta) &= p(a_{0:t}|o_{1:t}, \Theta) \\ &= \sum_{z_0, \dots, z_t=1}^{|\mathcal{Z}|} \left[ \mu(z_0) [\pi(z_0, a_0)] \prod_{\tau=1}^t W(z_{\tau-1}, a_{\tau-1}, o_\tau, z_\tau) [\pi(z_\tau, a_\tau)] \right] \end{aligned} \quad (21)$$

- $\mathcal{Z}$  is a finite set of decision regions
- $W(z, a, o', z') = p(z'|a, o', z)$  are decision-state transition matrices
- $\mu(z)$  is the initial distribution of decision states
- $\pi(z, a) = p(a|z)$  is state-dependent *stochastic* policies

# Regionalized Policy Representations (RPR)(Cont'd)

Our goal is to find the parameter  $\Theta = \{W, \mu, \pi\}$  to maximize the empirical value estimation  $\hat{V}(\mathcal{D}^{(K)}; \Theta)$

$$\hat{\Theta} = \arg \max_{\Theta} \frac{1}{K} \sum_{k=1}^K \sum_{t=0}^{T_k} \frac{\gamma^t r_t^k}{\prod_{\tau=0}^t p^{\Pi}(a_{\tau}^k | h_{\tau}^k)} \prod_{\tau=0}^t p(a_{\tau}^k | h_{\tau}^k, \Theta) \quad (22)$$

which is solved by maximum-value (MV) estimation or variational-Bayesian (VB) learning.

# Regionalized Policy Representations (RPR)(Cont'd)

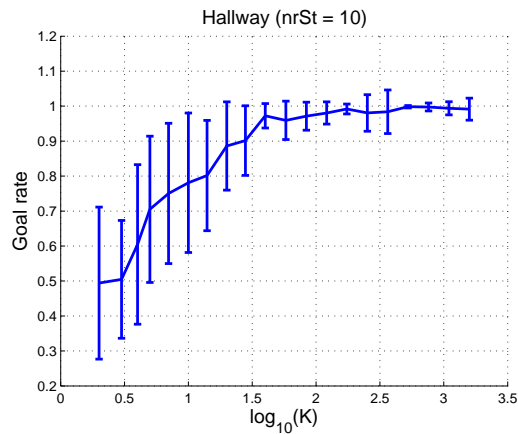
## Experimental Results

Table 4: A comparison of the RPR to other reinforcement learning algorithms on Hallway2

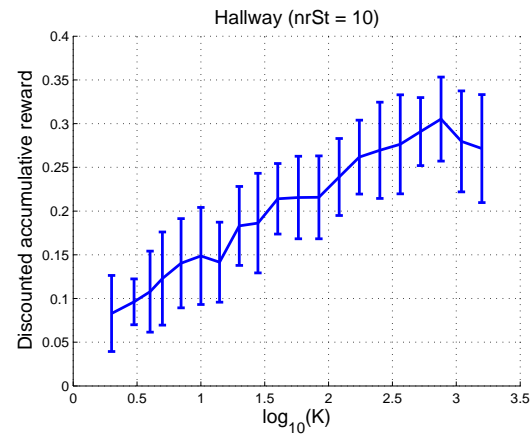
Method	Goal (%)	Median Steps
Random Walk	26	> 251
SARSA( $\lambda$ )(Loch and Singh 1998)	77	73
RL-LSTM (Bakker 2004)	94	61
UDHMM (Wierstra and Wiering 2004)	92	62
RPR (*)	97	46

# Regionalized Policy Representations (RPR)(Cont'd)

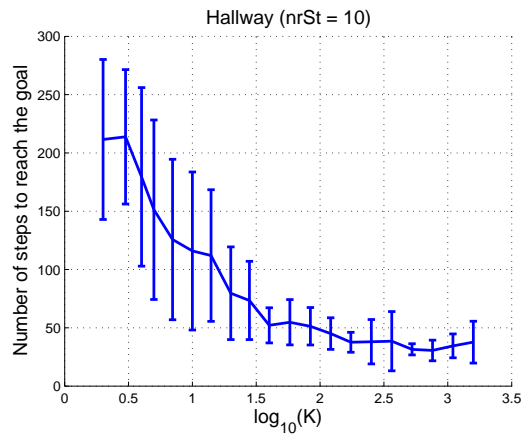
## Experimental Results



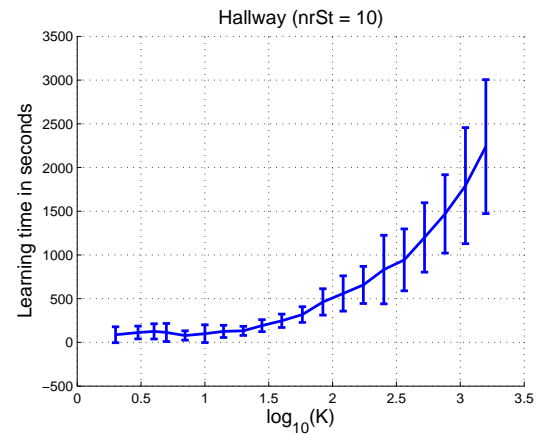
(a) Reward



(b) Goal rate



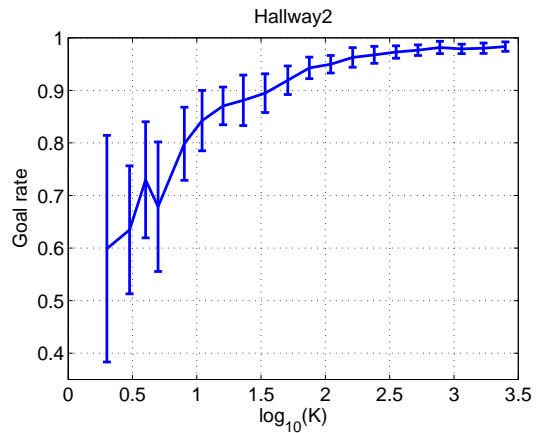
(c) Number of steps



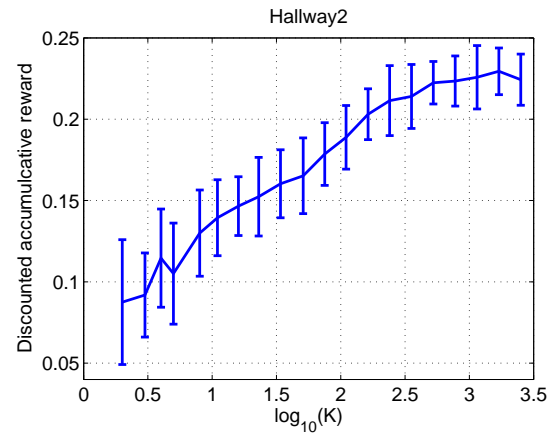
(d) Learning time

# Regionalized Policy Representations (RPR)(Cont'd)

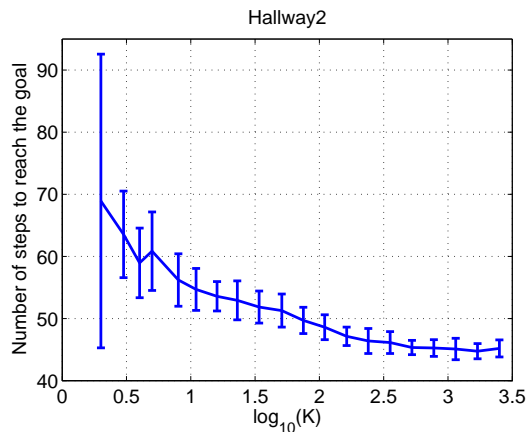
## Experimental Results



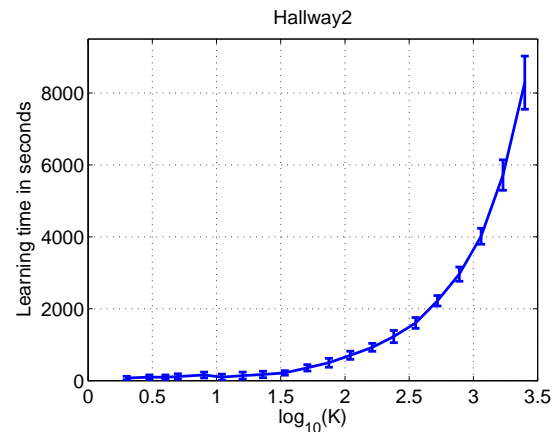
(e) Reward



(f) Goal rate



(g) Number of steps



(h) Learning time

# Regionalized Policy Representations (RPR)(Cont'd)

## Conclusions on the RPR

- We have presented the regionalized policy representation (RPR) to yield a history-dependent stochastic policy for environments characterized by a partially observable Markov decision process (POMDP).
- The RPR is learned in a model-free manner, based on a set of experiences collected through interaction with the environment.
- We have developed algorithms for learning the maximum-value RPR and the variational posterior PRP.
- The results show that the RPR is a powerful model-free policy representation that yields policies competitive with those of state-of-the-art algorithms.

# Outline

- Review of model-based POMDPs and policy design, motivation for RL
- Idea behind regionalized policy representation (RPR) for RL in POMDPs
- RPR implementation
- Multi-task and life-long learning with POMDPs
- Example results
- Future directions

# RL in Multiple POS Environments

- Motivation of the RPR for the multi-task learning (RPR-MTL)
- Idea of the RPR-MTL
- Algorithm of the RPR-MTL
- Implementation of the RPR-MTL
- Results of the RPR-MTL
- Conclusions of the RPR-MTL

# RL in Multiple POS Environments (Cont'd)

Motivation of the RPR for the multi-task learning (RPR-MTL)

- Learning policies for multiple partially observable stochastic (POS) environments
- The environments are not identical to each other
- The environments are not independent of each other

# RL in Multiple POS Environments (Cont'd)

## Problem

- How to find the common structure shared across multiple environments and how to represent the sharing structure

## Idea of the RPR-MTL

- We employ a hierarchical Bayesian approach, such that
  - Each environment has its own prior for the parameters  $\Theta = \{W, \mu, \pi\}$
  - The priors across all environments are drawn from the same Dirichlet process (DP)

# RL in Multiple POS Environments (Cont'd)

## Review of the Dirichlet Process (DP)

Let  $\mathcal{X}$  be a space. Let  $G_0$  be a base probability measure of  $X \in \mathcal{X}$  and  $\alpha$  be a positive scalar. Let  $B_1, B_2, \dots, B_k$  be a partition of  $\mathcal{X}$  satisfying  $B_i \cap B_j = \emptyset \forall i \neq j$  and  $\cup_{i=1}^k B_i = \mathcal{X}$ . A stochastic process  $G$  is called a Dirichlet process if, for any partition  $B_1, B_2, \dots, B_k$  of  $\mathcal{X}$ ,

$$\begin{aligned} & (G(X \in B_1), G(X \in B_2), \dots, G(X \in B_k)) \\ \sim & \text{Dir}(\cdot | \alpha G_0(X \in B_1), \alpha G_0(X \in B_2), \dots, \alpha G_0(X \in B_k)) \end{aligned} \quad (23)$$

# RL in Multiple POS Environments (Cont'd)

## Some properties of the DP

- The expectation of  $G$

$$E(G) = G_0 \quad (24)$$

- The posterior distribution of  $G$

$$p(G|X_1, X_2, \dots, X_n) = \text{DP} \left( \alpha + n, \frac{\alpha}{\alpha + n} G_0 + \frac{1}{\alpha + n} \sum_{i=1}^n \delta_{X_i} \right) \quad (25)$$

- The conditional probability of a new observation  $X_{n+1}$

$$p(X_{n+1}|X_1, X_2, \dots, X_n, \alpha, G_0) = \frac{\alpha}{\alpha + n} G_0 + \frac{1}{\alpha + n} \sum_{i=1}^n \delta_{X_i} \quad (26)$$

# RL in Multiple POS Environments (Cont'd)

## Algorithm of the RPR-MTL

- Considering  $m = 1, 2, \dots, M$  environments
- Each  $\Theta_m$  has its own prior

$$\begin{aligned}\Theta_m | G &\sim G \\ G | \alpha, G_0 &\sim DP(\alpha, G_0)\end{aligned}\tag{27}$$

- Polya urn scheme
  - Let  $\bar{\Theta} = \{\bar{\Theta}_n : n = 1, \dots, N\}$  be the set of distinct  $\Theta_m$ , where  $N \leq M$
  - Let  $c = \{c_1, c_2, \dots, c_M\}$  denote the vector of indicator variables defined by  $c_m = n$  iff  $\Theta_m = \bar{\Theta}_n$

# RL in Multiple POS Environments (Cont'd)

## Algorithm of the RPR-MTL

- Prior conditional distribution of the indicator

$$p(c_m | c_{-m}, \alpha) = \frac{\alpha}{\alpha + M - 1} \delta(c_m - N - 1) + \sum_{n=1}^N \frac{l_{-m,n}}{\alpha + M - 1} \delta(c_m - n) \quad (28)$$

- Posterior conditional distribution of the indicator

$$p(c_m | c_{-m}, \bar{\Theta}, \mathcal{D}_m^{(K_m)}, z^m, \alpha, G_0) = \beta_0^m \delta(c_m - N - 1) + \sum_{n=1}^N \beta_n^m \delta(c_m - n) \quad (29)$$

# RL in Multiple POS Environments (Cont'd)

## Algorithm of the RPR-MTL

$$\beta_0^m = \frac{\alpha C_{\hat{G}_0}^m}{\alpha C_{\hat{G}_0}^m + \sum_{n=1}^N l_{-m,n} \hat{V}(\mathcal{D}_m^{(K_m)}, z^m; \bar{\Theta}_n)} \quad (30)$$

$$\beta_j^m = \frac{l_{-m,j} \hat{V}(\mathcal{D}_m^{(K_m)}, z^m; \bar{\Theta}_j)}{\alpha C_{\hat{G}_0}^m + \sum_{n=1}^N l_{-m,n} \hat{V}(\mathcal{D}_m^{(K_m)}, z^m; \bar{\Theta}_n)} \quad (31)$$

$$C_{\hat{G}_0}^m = \int \hat{V}(\mathcal{D}_m^{(K_m)}, z^m; \Theta_m) G_0(\Theta_m) d\Theta_m \quad (32)$$

- When the  $n$ -th distinct RPR produces a high empirical value in the  $m$ -th environment,  $c_m$  tends to equal  $n$
- Otherwise,  $c_m$  tends to equal  $N + 1$ , which means a new  $\Theta$  will be drawn from the base  $G_0$ .

# RL in Multiple POS Environments (Cont'd)

## Algorithm of the RPR-MTL

Once the samples of the indicator variables are given, we put together the episodes from the environments whose indication variables are equal, and draw  $\bar{\Theta}_n$  according to

$$\begin{aligned} p(\bar{\Theta}_n | c, \mathcal{D}, z) &= \frac{\sum_{m \in I_n(c)} V(\mathcal{D}_m^{(K_m)}, z^m; \bar{\Theta}_n) G_0(\bar{\Theta}_n)}{\int \sum_{m \in I_n(c)} V(\mathcal{D}_m^{(K_m)}, z^m; \bar{\Theta}_n) G_0(\bar{\Theta}_n) d\bar{\Theta}_n} \\ &= \sum_{m \in I_n(c)} \left[ \sum_{k=1}^{K_m} \sum_{t=0}^{T_{m,k}} \tilde{\beta}_0^{m,k,t} \hat{G}_0^{m,k,t}(\bar{\Theta}_n) \right] \end{aligned} \quad (33)$$

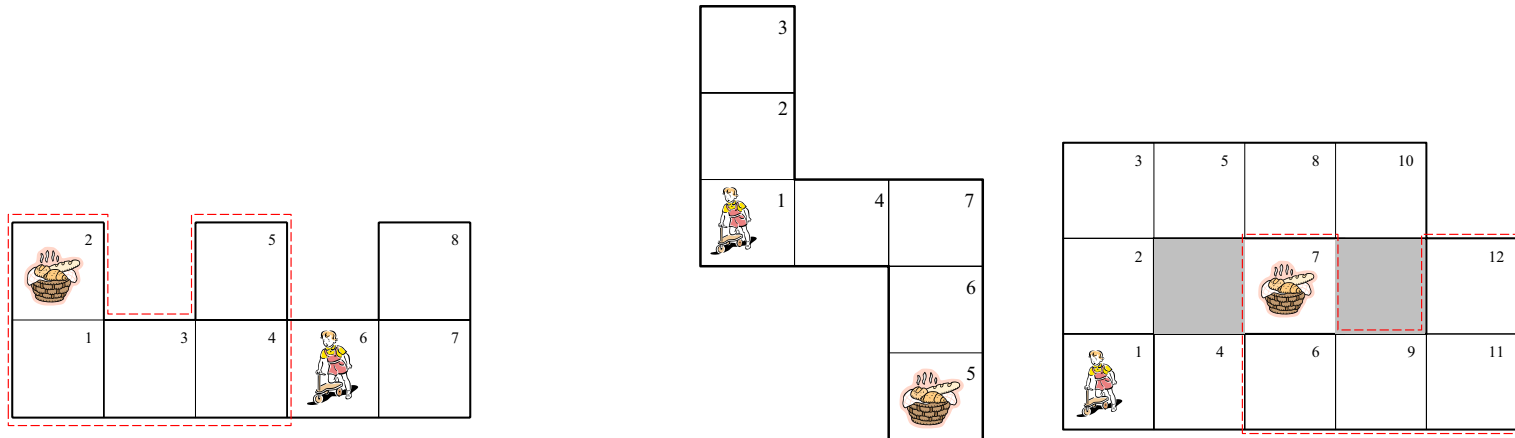
# RL in Multiple POS Environments (Cont'd)

## Implementation of the RPR-MTL

- 1: Initialization:  $N = M$ ,  $c_m = m$  for  $m = 1, 2, \dots, M$ , initialize  $\{\bar{\Theta}_n\}_{n=1}^N$ .
- 2: Draw hidden decision states or update the posterior distribution of hidden decision states
- 3: Draw indicator variables according to (29)
- 4: Draw indicator variables according to (33); go back to step 2 until convergence

# RL in Multiple POS Environments (Cont'd)

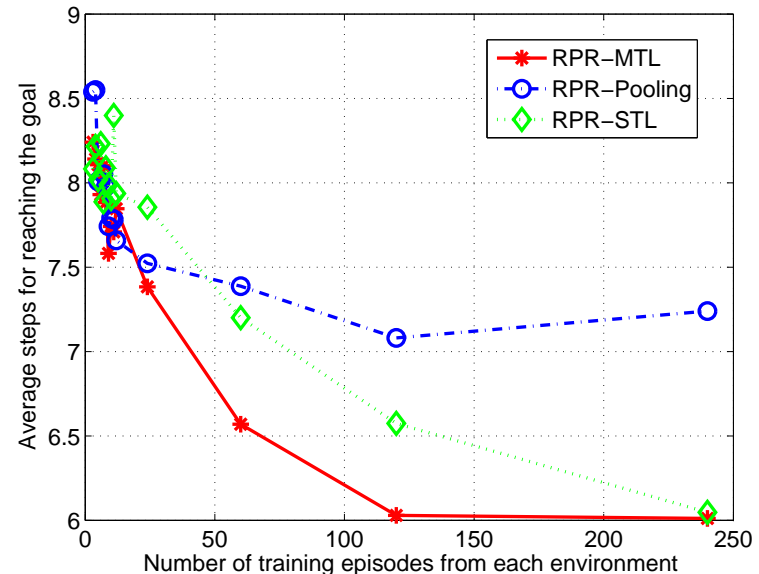
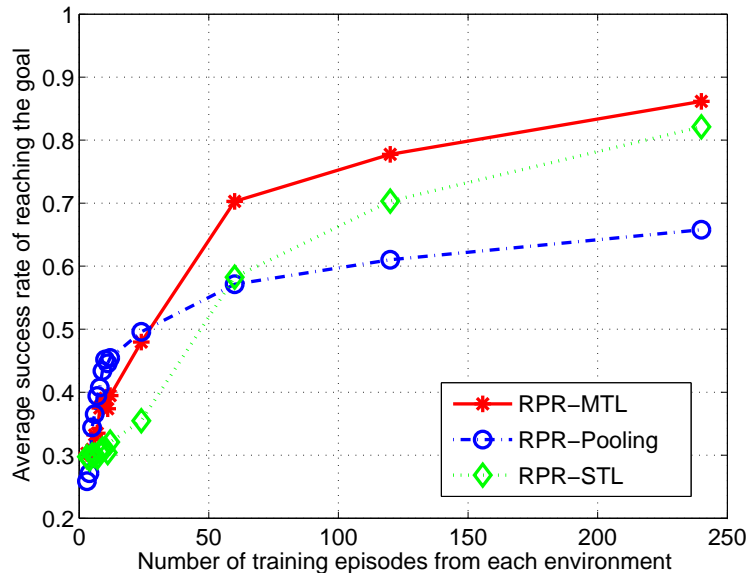
Experimental results on multiple POS environments



- Ten environments, of which there are three distinct environments
- The first three are from the first distinct environments
- The following three are from the second distinct environments
- The last four from the third distinct environments

# RL in Multiple POS Environments (Cont'd)

## Experimental results on multiple POS environments

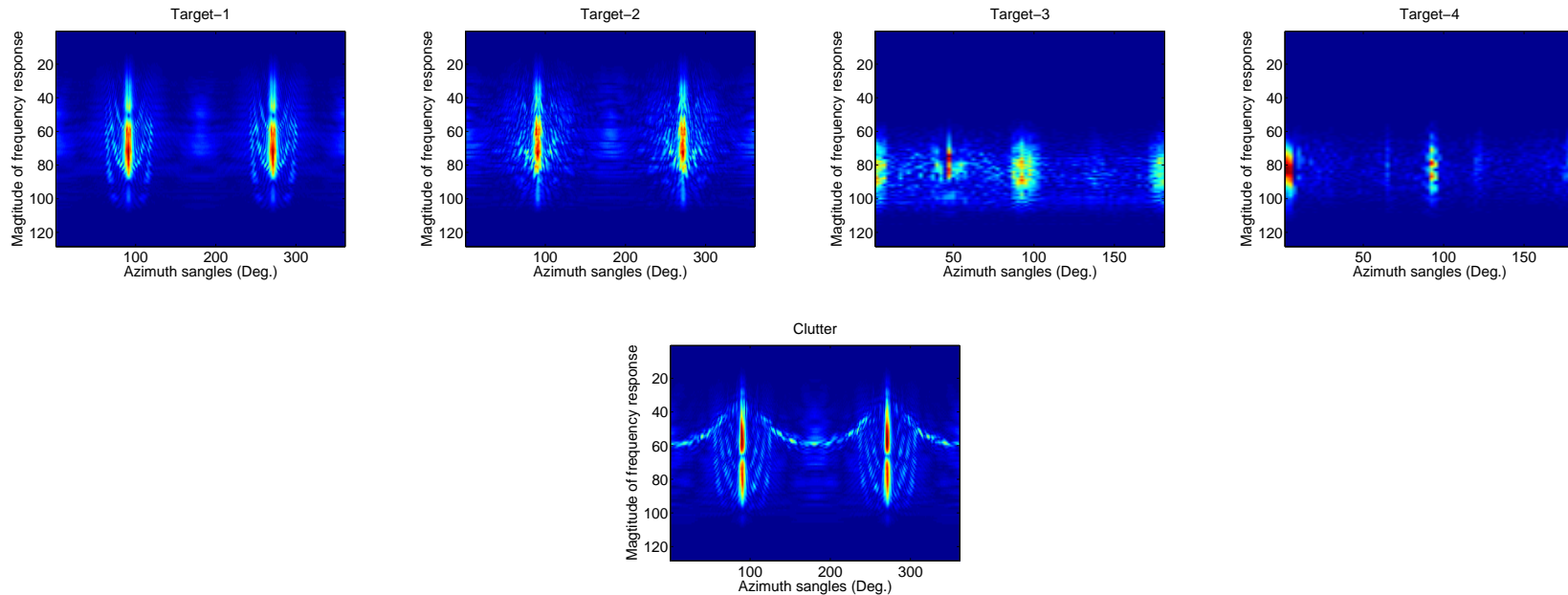


- When training episodes are scarce, pooling is good.
- When episodes increase, pooling them together is bad.
- The RPR-MTL algorithm automatically finds out appropriate sharing among the environments to improve the overall performance.



# RL in Multiple POS Environments (Cont'd)

## Experimental results on multi-aspect target classification

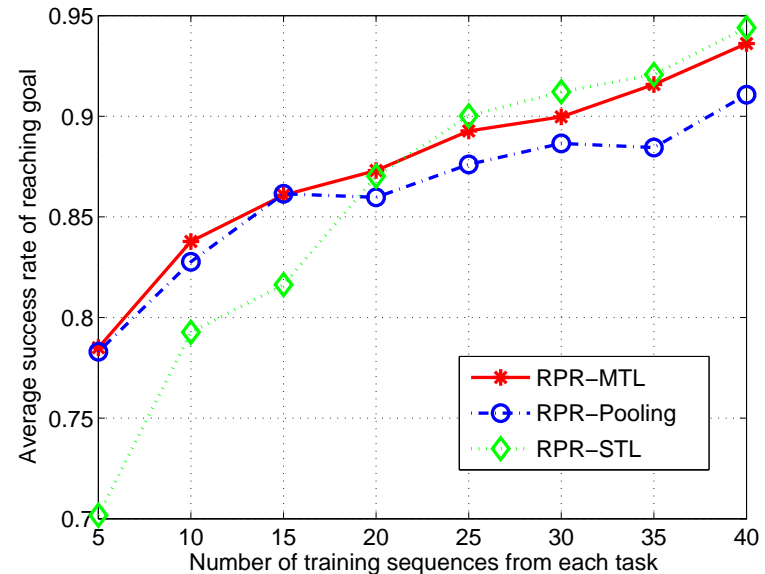
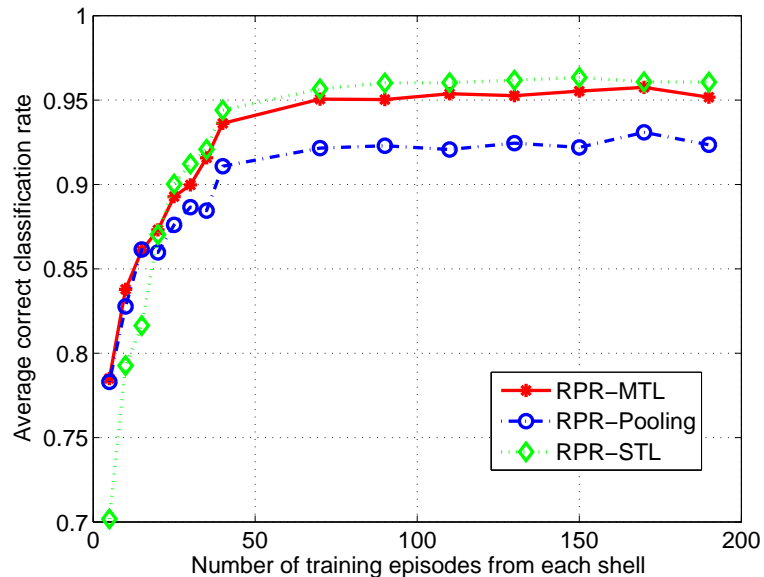


### Four tasks:

- Classify Target 1 and clutter
- Classify Target 2 and clutter
- Classify Target 3 and clutter
- Classify Target 4 and clutter

# RL in Multiple POS Environments (Cont'd)

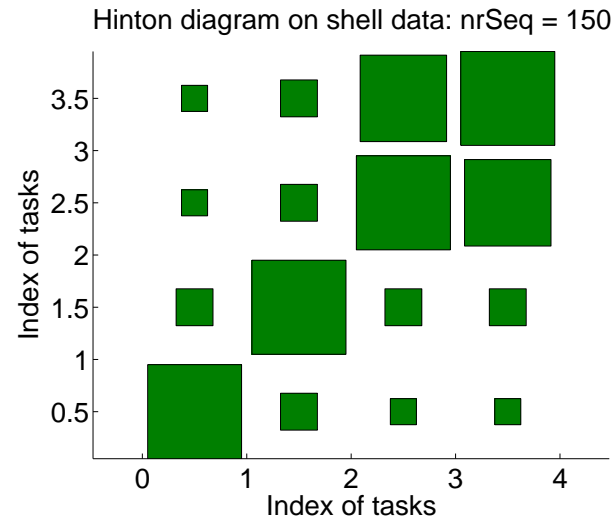
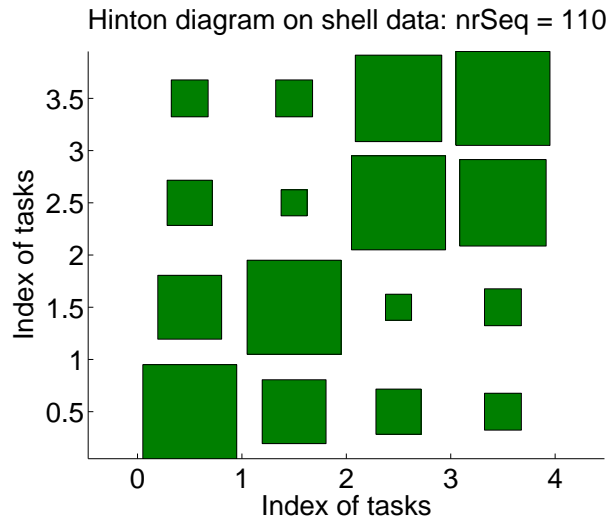
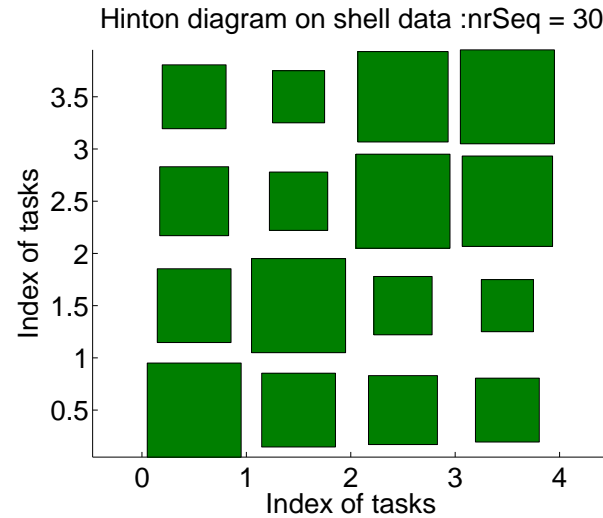
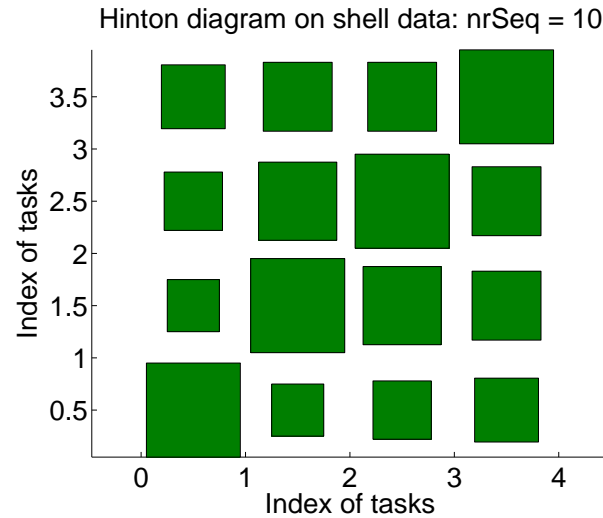
## Experimental results on multi-aspect target classification



- RPR-MTL achieves a performance comparable to the best between the RPR-STL and the RPR-Pooling.
- RPR-MTL adaptively adjusts the sharing among tasks as the number of training episodes changes, such that the sharing is appropriate regardless of the number of episodes.

# RL in Multiple POS Environments (Cont'd)

## Sharing patterns on multi-aspect target classification



# RL in Multiple POS Environments (Cont'd)

## Conclusions on the RPR-MTL

- we have extended the regionalized policy representation (RPR) to multitask learning, where the RPRs for multiple environments are learned simultaneously under a unified framework.
- The sharing is achieved by placing a Dirichlet process (DP) prior on the RPRs across all environments.
- The multitask-RPR is demonstrated by experimental results, which show that the multitask-RPR can constantly perform the best regardless of the number of episodes.

# Outline

- Review of model-based POMDPs and policy design, motivation for RL
- Idea behind regionalized policy representation (RPR) for RL in POMDPs
- RPR implementation
- Multi-task and life-long learning with POMDPs
- Example results
- Future directions

# Summary & Future Directions

- RPR RL algorithm goes directly from experiences to policy
- HMM-like stochastic policy representation compact and easy to use
- Multi-task and life-long learning share previous experiences appropriately
- Removes the (often very strong) assumption that the POMDP model is known, no longer explicitly computing belief states
- Since policy is based directly on experience, well poised to address joint exploration and exploitation
- To date joint exploration & exploitation an unsolved problem for POMDPs