Bayesian Nonparametric Policy Learning in Dec-POMDPs

Miao Liu¹, Chris Amato², Xuejun Liao³, Lawrence Carin³, Jonathan P. How¹

¹Massachusetts Institute of Technology, ²University of New Hampshire, ³Duke University

December 12th 2015

MALIC15 NIPS Talk/ Main work was publised in IJCAI15
Motivation

- Decentralized control enables new missions for autonomous systems while reducing burden on manpower/communication infrastructure
- Decentralized Partially Observable Markov Decision Processes (Dec-POMDPs): Extend single agent MDP and POMDP models
  - General representation of multi-agent planning under uncertainty problem

- Search and rescue
- UAV surveillance
- Cyber security
- Space exploration
- Warehousing
- Human-robot teaming
Multiagent Autonomy Challenges

▶ **Challenge:** Difficult to generate detailed models of all the agents in these complex, real-world settings

▶ **This paper:** a full Bayesian method to optimize policies directly from data, without the need of such models
  - Each agent’s policy represented as a **Finite State Controller (FSC)**
  - Number of nodes in each FSC is **learned from data**
  - Efficient **fully Bayesian inference** via variational Bayes
Decentralized Partially Observable Markov Decision Processes (DEC-POMDPs)

$M_{DEC-POMDP}(\mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{O}, T, \Omega, R)$

- $\mathcal{N} = \{1, \ldots, N\}$ is a finite set of agents
- $\mathcal{A}_n$, each agent’s finite set of actions
- $\mathcal{O}_n$, each agent’s finite set of observations
- $\mathcal{S}$ is a set of states for the system
- $T$, the state transition model: $Pr(s'|s, \bar{a})$
- $\Omega$, the observation model: $Pr(\bar{o}|s', \bar{a})$
- $R$, the reward model: $r(s, \bar{a})$

Goal: maximize expected cumulative reward

A solution to an infinite horizon Dec-POMDP is a joint policy, a set of **finite state controller (FSC)**, one for each agent

- $\mathcal{Z}_n$ a finite set of (decision) nodes: abstraction of history
- $W_n$, controller nodes transition model, $W^{zao}_{n,z'} = Pr(z'|z, a, o, n)$
- $\pi_n$ a set of local policies, $\pi^{a}_{n,z} = Pr(a|z, n)$
Introduction

Stick-Breaking Policy Learning in DEC-POMDPs

- **Usual Assumption**: Detailed models of all the agents in the Dec-POMDP are available
- **Problem**: What if we don’t have access to that?
- **Solution**: Use previously executed action, observation histories $H$, and rewards to improve future decision making
- **Prior State-of-the-art**: Uses fixed-sized finite state controllers to represent each agent’s policy, and uses EM for inference: leads to under/over-fitting issues [Wu et al. IJCAI13]
- **Result**: Novel approach to learn policies from data called DecSBPR (decentralized stick-breaking policy representation)
  - Automatically learns number of FSC states for each agent from data
  - Off-policy learning is available by experimentation in real mission domains
Empirical value function: $\hat{V}(D^K; \Theta)$ is constructed by data collected by following some behavior policies
- $\hat{V}(D^K; \Theta)$ can be converted into a likelihood function $L(D^K; \Theta)$

The posterior of the joint FSC: $p(\Theta|D^K) \propto L(D^K; \Theta)p(\Theta)$
- FSC node transition probabilities model via stick-breaking priors

An illustration of FSCs
The decentralized stick-breaking policy representation (Dec-SBPR): \((Z, \mu, \eta, \rho)\)

- \(Z = \bigotimes Z_n\) is an unbounded set of decision states
- \((\eta, \rho)\) determine \((W, \pi)\), the FSC parameters

\[
W^{i,1:1:}\infty_{n,a,o} \sim \text{SB}(\sigma^{i,1:1:}\infty_{n,a,o}, \eta^{i,1:1:}\infty_{n,a,o}), \quad \pi^{1:|A_n|}_{n,i} \sim \text{Dir}(\rho^{1:|A_n|}_{n,i}),
\]

\[
\eta^{i;j}_{n,a,o} \sim \text{Gamma}(c, d), \quad n = 1, \ldots, N, i, j = 1, \ldots, \infty
\]

- \(\text{SB}\) represents stick-breaking prior with
  
  \[
  W^{i,j}_{n,a,o} = V^{i,j}_{n,a,o} \prod_{m=1}^{j-1} (1 - V^{i,m}_{n,a,o})
  \]
  
  and \(V^{i,j}_{n,a,o} \sim \text{Beta}(\sigma^{i,j}_{n,a,o}, \eta^{i,j}_{n,a,o})\)

- \(\eta\) controls the sparsity of \(W\)
- \(\text{Dir}\) represents Dirichlet distribution

An illustration of the Stick-breaking process.

---

\[
\begin{align*}
W^{i,1}_{n,a,o} & \quad \cdots \quad W^{i,\infty}_{n,a,o} \\
V^{i,1}_{n,a,o} & \quad 1 - V^{i,1}_{n,a,o} \\
V^{i,2}_{n,a,o} (1 - V^{i,1}_{n,a,o}) & \quad (1 - V^{i,2}_{n,a,o})(1 - V^{i,1}_{n,a,o}) \\
& \quad \vdots
\end{align*}
\]

Recursively break the stick of length 1
Variational Bayesian Inference of Decentralized Policies

- Denote \( q(\Theta)q^k_t(\tilde{z}^{k}_{0:t}) \) as the variational approx. to \( p(\Theta, \tilde{z}^{k}_{0:t}|\mathcal{D}(K)) \)
- Learning decentralized policies as a constrained optimization problem

\[
\max_{\{q^k_t(\tilde{z}^{k}_{0:t})\}} \text{LB}(\{q^k_t(\tilde{z}^{k}_{0:t})\}, q(\Theta))
\]

subject to: \( q^k_t(\tilde{z}^{k}_{0:t}, \Theta) = \prod_{n=1}^{N} q^k_t(z^{k}_{n,0:t})q(\Theta_n) \Rightarrow \) mean-field approx. & decentralized policy representations

\[
\sum_{k=1}^{K} \sum_{t=0}^{T_k} \sum_{z^{k}_{1:N,0:t}=1}^{|\mathcal{Z}|} q^k_t(\tilde{z}^{k}_{0:t}) = K, \quad q^k_t(\tilde{z}^{k}_{0:t}) \geq 0, \quad \forall \tilde{z}^{k}, t, k,
\]

\[
\int p(\Theta)\,d\Theta = 1 \quad \text{and} \quad p(\Theta) \geq 0, \quad \forall \Theta
\]

\( \Rightarrow \) normalization constraints
Experimental Results

DEC-SBPR Results

► Learning variable vs fixed-size FSCs
  ▪ EM algorithm suffers from over/under-fitting issue
  ▪ Using the stick-breaking (SB) prior achieves slightly better performance than the Dirichlet prior, which can be explained the flexibility of SB prior

![Graph showing policy value vs number of nodes for Mars Rover, Recycle Robots, Box Pushing, and Mars Rovers]

Batch sequential learning and comparison with other methods
  ▪ DEC-SBPR achieves better results to the state-of-art policy-based method MCEM [Wu et al, IJCAI13]
  ▪ Periodic EM (PeriEM) [Pajarinen&Peltonen, NIPS11] and FB-HSIVI [Dibangoye et al, 14], two state-of-art planning methods (with known models) are treated as supper-bounds for the policy-based methods

| Problems (|S|, |A|, |O|)            | POLICY LEARNING (UNKNOWN MODEL) | PLANNING (KNOWN MODEL) |
|--------------|----------------|------------------|------------------|
|              | Dec-SBPR (fixed iteration) | Dec-SBPR (fixed time) | MCEM |          | PeriEM |          | FB-HSIVI |          |
|              | Value | Z | Time | Value | Z | Time | Value | Z | Time | Value | Z | Time |
| Broadcast (4, 2, 5) | 9.20 | 2 | 7s | 9.27 | 2 | 24s | 9.15 | 3 | 24s | - | - | 9.27 | 102 | 19.8s |
| Recycling Robots (3, 3, 2) | 31.26 | 3 | 147s | 25.16 | 2 | 19s | 30.78 | 3 | 19s | 31.80 | 6 * 10 | 272s | 31.93 | 108 | 0s |
| Box Pushing (100, 4, 5) | 77.65 | 14 | 290s | 58.27 | 9 | 32s | 59.95 | 3 | 32s | 106.68 | 4 * 10 | 7164s | 244.43 | 331 | 1715.1s |
| Mars Rovers (256, 6, 8) | 20.62 | 5 | 1286s | 15.2 | 6 | 160s | 8.16 | 3 | 160s | 18.13 | 3 * 10 | 7132s | 26.94 | 136 | 74.31s |

NIPS15 Workshop (MIT,UNH,Duke) Bayesian Nonparametric Policy Learning in Dec-POMDPs
Experimental Results

DEC-SBPR Results

- Learning variable vs fixed-size FSCs
  - EM algorithm suffers from over/under-fitting issue
  - Using the stick-breaking (SB) prior achieves slightly better performance than the Dirichlet prior, which can be explained the flexibility of SB prior

- Batch sequential learning and comparison with other methods
  - DEC-SBPR achieves better results to the state-of-art policy-based method MCEM [Wu et al, IJCAI13]
  - Periodic EM (PeriEM) [Pajarinen&Peltonen, NIPS11] and FB-HSVI [Dibangoye et al, 14], two state-of-art planning methods (with known models) are treated as supper-bounds for the policy-based methods
Exploration and Exploitation in Sequential 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]
  - \[ \sigma_{y}^{z} = p(y = 0|z) \]

\[ \sigma_0^z \sim \text{Beta}(u_0^z, u_1^z), \text{ with } \sigma_1^z = 1 - \sigma_0^z, \forall z \in \mathcal{Z}, \]  
(3)

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

\[ u_0^i = \sum_{k=1}^{K} \sum_{t=0}^{T_k} \hat{\nu}_t^k \sum_{\tau=0}^{t} \phi_{t,\tau}^k(i), \forall i \in \mathcal{Z}, \]  
(4)

- **\( u_1 \)** 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).
Plots for illustrating exploration-exploitation tradeoff, including testing value (left), inferred controller numbers (middle) and exploration rate (right).
Scaling up

Performance on the traffic control domain with $10^{20}$ states and 100 agents. Left: Domain illustration; Middle: test reward; Right: Inferred decision state number.

- Domain challenge
  - 100/2500 agent controlling traffic flow at $10 \times 10/50 \times 50$ intersections
  - $10^{20}/10^{100}$ states & infinite planning horizon
- Perform sequential batch policy learning with exploration and exploitation tradeoff
- Dec-SBPR infers controller node size ($|Z|$) and achieves better performance than EM with arbitrarily selected $|Z|$
Toward Realword Problems (To appear in AAAI16)

![Diagram](image1)

![Diagram](image2)

![Diagram](image3)

![Diagram](image4)
Summary

- Developed a scalable Bayesian nonparametric policy learning framework for solving Dec-POMDPs
  - without the need to know the full Dec-POMDP model a priori
  - scalable to large problem sizes and numbers of agents
  - allows inferring variable-size controllers
  - allows encoding the experts/prior knowledge
  - provides high-quality solutions from a small amount of data
  - outperforms other model-free methods

The future work
- Customizing Dec-SBPR for more realistic problems
- Extension to multi-agent (decentralized) learning