Large-Scale Markov Decision Problems with KL Control Cost and their Application to Crowdsourcing

Yasin Abbasi-Yadkori [1], Peter Bartlett [1,2], Xi Chen [3], **Alan Malek**[2]

[1]Queensland University of Technology

[2]University of California, Berkeley

[3]NYU Stern School of Business

July 7th, 2015

Conclusion

- Problem: MDP planning problem with large state space
- Goal: find near-optimal policy in low dimensional family of policies
- Novel framework for linearly solvable MDPs
- Also: Algorithm with complexity that scales with dimension of family
- First theoretical bounds for approximate solutions in linearly solvable MDPs
- Demonstrate on pratical example

Previous work

- Approximate Dynamic Programming (linear approximation of the value function): [Sutton and Barto, 1998, Bertsekas, 2007]
- Approximate Linear Programming: (approximately solving LP) [Schweitzer and Seidmann, 1985, de Farias and Van Roy, 2003, 2004, 2006, Hauskrecht and Kveton, 2003, Guestrin et al., 2004, Petrik and Zilberstein, 2009, Desai et al., 2012, Veatch, 2013].
- Solving LMDPs (with no theoretical guarantees): [Todorov, 2009] and [Zhong and Todorov, 2011a,b]
- Approximate policy iteration (e.g. least squares policy iteration)

Large Scale MDPs

- Markov decision process: modeling sequential decisions
- E.g. queueing network, robot planning
- Can solve for small state spaces
- Applications have *large* state spaces

Notation

A Markov Decision Process is specified by:

- State space $\mathcal{X} = \{1, \ldots, X\}$
- Action space $\mathcal{A}$
- Transition Kernel $K : \mathcal{X} \times \mathcal{A} \to \triangle_{\mathcal{X}}$
- Loss function $\ell : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^+$

Problem:

- Policy $\pi : \mathcal{X} \to \triangle_{\mathcal{A}}$
- Find policy to minimize value function

$$\boldsymbol{J}_\pi(x) = \mathbb{E}\left[\sum_{t=0}^{\infty} \ell(X_t, \pi) \middle| X_0 = x\right]$$

Aim for optimality *within a restricted family of policies.*

Large state space

- Parametric class of value functions $\boldsymbol{J}_\theta$ for $\theta \in \Theta \subset \mathbb{R}^d$
- Bellman operator:

$$(\boldsymbol{L}\boldsymbol{J})(x) = \min_{a \in \mathcal{A}} \left\{ \ell(x, a) + \mathbb{E}_{x' \sim P_0(x,a)} \boldsymbol{J}(x') \right\}$$

- Optimal policy $\boldsymbol{J}^*$ is a fixed point: $\boldsymbol{L}\boldsymbol{J}^* = \boldsymbol{J}^*$
- Greedy policy: $\pi_{\boldsymbol{J}_\theta}$ (the argmin)
- Ultimate goal: find a θ to minimize

$$\boldsymbol{J}_{\pi_{\boldsymbol{J}_\theta}},$$

  the actual value of the greedy policy of the approximate optimal value

Approximate solutions

- Consider the unconstrained surrogate

$$\min_{\theta} c^\top \boldsymbol{J}_\theta + \|\boldsymbol{L}\boldsymbol{J}_\theta - \boldsymbol{J}_\theta\|$$

- Can we solve this with algorithms that scale with d but not X?

KL-cost

- Introduced in [Todorov, 2006]
- $\mathcal{A} = \triangle_{\mathcal{X}}$
- Loss: $\ell(x, P) = q(x) + D_{KL}(P||P_0(\cdot|x))$
  - state loss $q(x)$, base dynamics P_0
  - infinite loss unless $P \ll P_0$
- Terminal state z
- Total cost of policy P

$$\boldsymbol{J}_P(x) = \mathbb{E}\left[\sum_{t=0}^{\infty} \ell(X_t, P) \middle| X_0 = x\right]$$

Linearly Solvable

- Greedy action is:

$$P_{\boldsymbol{J}}(\cdot|x) = \arg\min_{P \in \triangle_{\mathcal{X}}} \mathbb{E}_{y \sim P(\cdot|x)}[q(y) + \boldsymbol{J}_P(y)] \propto P_0(\cdot|x) e^{-\boldsymbol{J}_P(\cdot)}$$

- Bellman's operator becomes linear in $g(x) = e^{-\boldsymbol{J}(x)}$:

$$e^{-\boldsymbol{L}\boldsymbol{J}(x)} = e^{-q(x)} \sum_{x'} P_0(x, x') e^{-\boldsymbol{J}(x')}$$

- Bellman's optimality equation:

$$\boldsymbol{L}\boldsymbol{J} = \boldsymbol{J} \Leftrightarrow e^{-q} P_0 e^{-\boldsymbol{J}} = e^{-\boldsymbol{J}}$$

Parameterizing J_θ

- Previous ADP techniques used $J_\theta = \Psi\theta$
- Intuition: take $J_\theta = -\log(\Psi\theta)$ so e^{-LJ_θ} is linear in θ
- Surrogate optimization:

$$\min_\theta c^\top J_\theta + \underbrace{\|LJ_\theta - J_\theta\|}_{\text{Bellman error}} \tag{1}$$

- $\|LJ_\theta - J_\theta\|$ not convex in θ, but

$$e^{-\max\{LJ_\theta, J_\theta\}} \|LJ_\theta - J_\theta\| \leq \left\| e^{-LJ_\theta} - e^{-J_\theta} \right\|$$

- Plugging $\Psi\theta = e^{-J\theta}$ into (1):

$$\min_\theta -c^\top \log(\Psi\theta) + \| e^{-q} P_0 \Psi\theta - \Psi\theta \|$$

Parameterizing J_θ

- Previous ADP techniques used $J_\theta = \Psi\theta$
- Intuition: take $J_\theta = -\log(\Psi\theta)$ so e^{-LJ_θ} is linear in θ
- Surrogate optimization:

$$\min_\theta c^\top J_\theta + \underbrace{\|LJ_\theta - J_\theta\|}_{\text{Bellman error}} \tag{1}$$

- $\|LJ_\theta - J_\theta\|$ not convex in θ, but

$$e^{-\max\{LJ_\theta, J_\theta\}} \|LJ_\theta - J_\theta\| \leq \left\|e^{-LJ_\theta} - e^{-J_\theta}\right\|$$

- Plugging $\Psi\theta = e^{-J\theta}$ into (1):

$$\min_\theta -c^\top \log(\Psi\theta) + \|\underbrace{e^{-q}P_0\Psi\theta}_{\substack{\text{Bellman}\\\text{operator}}} - \Psi\theta\|$$

Our algorithm

- Recall relaxed optimization:

$$\min_{\theta} -c^\top \log(\Psi\theta) + \left\| e^{-q} P_0 \Psi\theta - \Psi\theta \right\|_Q$$

- Let $\mathcal{T}$ be the set of trajectories with $x_1 \sim c$ with distribution $Q(\cdot)$
- Optimization is equal to:

$$\min_{\theta} -c^\top \log(\Psi\theta) + \sum_{\boldsymbol{T} \in \mathcal{T}} Q(\boldsymbol{T}) \sum_{x \in \mathcal{T}} \left| e^{-q(x)} P_0 \Psi\theta(x) - \Psi\theta(x) \right|$$

- Use stochastic gradient descent by sampling trajectories

Theorem

Let $\widehat{\theta}$ be an ϵ-optimal solution returned by SGD. Then,

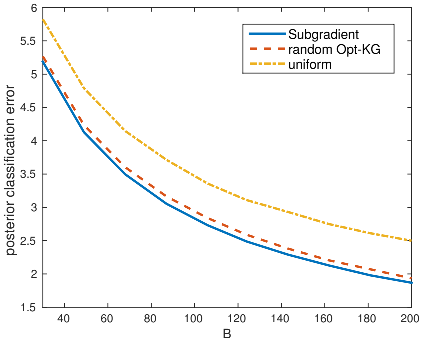
$$J_{P_{J_{\widehat{\theta}}}}(x_1) \le \inf_{\theta \in \Theta} \left\{ J_{P_{J_\theta}}(x_1) + \mathcal{E}(J_\theta) \right\} + \epsilon$$
$$+ \underbrace{\left\| P_{J_{\widehat{\theta}}} - Q \right\|_1}_{\text{Off-policy error}} \max_{T \in \mathcal{T}} \sum_{x \in T} \left| J_{\widehat{\theta}}(x) - L J_{\widehat{\theta}}(x) \right|$$

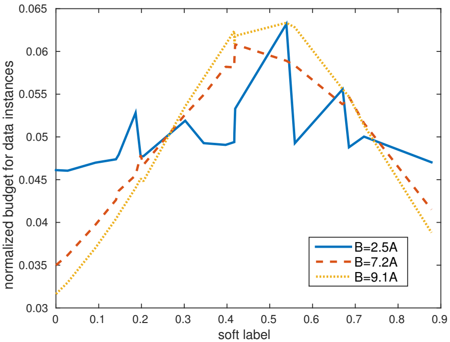
Penalty function:

$$\mathcal{E}(J_\theta) = \sum_{T \in \mathcal{T}} \sum_{x \in T} \left(Q(T) e^{-\min(J_\theta, L J_\theta)} + P_{J_\theta}(T) \right) \underbrace{\left| J_\theta(x) - L J_\theta(x) \right|}_{\text{Small if } J_\theta \text{ is close to the optimal value}}$$

Crowdsourcing

- Need to label A items.
- Each item has soft label $\mu_i \in [0, 1]$
- Guess if $\mu_i \geq \frac{1}{2}$ for as many i as we can
- For $t = 1, \ldots, T$:
  - Pick $i \in \{1, \ldots, A\}$
  - Receive $X_t \sim \text{Bern}(\mu_i)$
- Use Beta prior $\Rightarrow$ MDP dynamics equivalent to Bayesian updates
- P_0 limits transitions
- $q(x)$ rewards correct labels





- Average error of three policies
- Our method requires 10% fewer samples for same accuracy

- Portion of budget vs. soft label
- Harder soft labels receive more budget
- Larger difference as B grows

Conclusion

- Novel framework for low dimensional policies for linearly solvable MDPs
- Algorithm for policy optimization with complexity that scales with dimension of subspace
- First theoretical bounds for approximate linearly solvable MDP solutions
- Demonstrate on pratical example

Thanks!

Proof outline of main theorem

- $\left| \boldsymbol{J}_{P_{\boldsymbol{J}_{\theta^*}}}(x_1) - \boldsymbol{J}_{\theta^*}(x_1) \right| = O\left(\| \boldsymbol{L}\boldsymbol{J}_{\theta^*} - \boldsymbol{J}_{\theta^*} \| \right)$
- Similarly bounding $\left| \boldsymbol{J}_{P_{\boldsymbol{J}_{\widehat{\theta}}}}(x_1) - \boldsymbol{J}_{\widehat{\theta}}(x_1) \right| = O\left(\left\| \boldsymbol{L}\boldsymbol{J}_{\widehat{\theta}} - \boldsymbol{J}_{\widehat{\theta}} \right\| \right)$
- $\boldsymbol{J}_{\theta^*}$ and $\boldsymbol{J}_{\widehat{\theta}}$ are close by the optimization