Markov Decision Processes with Large State Space

Ojas Deshpande   Chirag Maheshwari

May 9, 2017

Department of Computer Science
New York University

Table of contents

Introduction

Markov Decision Process

Markov Decision Process (MDP) is a stochastic system defined by a tuple $\mathcal{M} =< \mathcal{X}, \mathcal{A}, \mathcal{P}, R, \gamma >$ where,

- $\mathcal{X}$ is a countable set of states (state space).
- $\mathcal{A}$ is a countable set of actions (action space).
- $P \in \mathcal{P}$, $P : \mathcal{X} \times \mathcal{A} \to \Delta_{\mathcal{X}}$ is the probability transition matrix between states given an action.
- $R : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ is an immediate reward (cost) function.
- $\gamma \in [0, 1]$ is the discount factor.

Control Policy

A policy $\pi : \mathcal{X} \to \Delta_{\mathcal{A}}$ denotes the probability to choose action $a \in \mathcal{A}$ given state $x \in \mathcal{X}$.

Each policy induces a transition matrix P^π

$$P^\pi(x, x') = \sum_{a \in \mathcal{A}} p(x'|x, a)\pi(x, a)$$

Stationary Distribution of states seen under policy π is denoted by v_π. Stationary Distribution over state-action space can be defined as,

$$\mu_\pi(x, a) = v_\pi(x)\pi(x, a) \Rightarrow \pi(a|x) = \frac{\mu(x, a)}{\sum_{a' \in \mathcal{A}} \mu(x, a)}$$

Aim is to *device* an efficient policy to maximize expected reward (minimize expected cost).

Total Costs

Starting from some initial state X_1, *total cost* is defined as:

$$h(x_1) = \lim_{T \to \infty} \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} l(x_t, P^{\pi})\right]$$

Also called the *value function*.

Note: for our purposes we assume $\gamma = 1$.

To make the total cost well defined, we have a set of *absorbing states* $S \subset \mathcal{X}$ such that, $l(s, P) = 0$ and $P(s, s) = 1$.

Average Costs

Starting from some initial state X_1, *average cost* a.k.a. ergodic cost is defined as:

$$J(x_1) = \lim_{T \to \infty} \frac{1}{T} \mathbb{E}\left[\sum_{t=1}^{T} l(x_t, P^{\pi})\right]$$

As there exists a Stationary Distribution over the states (property of *ergodicity*) this average cost is well defined and is independent of the starting state. Thus,

$$J(x) = \lambda \ \forall x \in \mathcal{X}$$

The *differential value function* denoted by $h(x)$ is the difference between actual cost and average cost.

Applications

MDPs can be used to model many real-life problems

- Resource Allocation
- Queue Control
- Routing
- Inventory Control
- Robotics
- Games
- Asset Pricing
- Risk Management
- Power Grid Management
- Crowd-sourcing Budget Allocation
- Sequential Clinical Trials
- Scheduling systems

Problem

Bellman Optimality Equation

Bellman optimality operator defined by,

$$(Lh)(x) = \min_{a \in \mathcal{A}} \left(l(x,a) + \sum_{x' \in \mathcal{X}} P_{(x,a),x'} \right) h$$

gives the bellman optimality equation [5],

$$\lambda_* + h_*(x) = (Lh_*)(x)$$

- $\lambda = 0$ in case of Total Cost problems
- Solving an MDP is computationally intensive and is *P-complete*.
- Policy Iteration and Value Iteration has $O(|\mathcal{X}|^2|\mathcal{A}|)$ per-iteration complexity.

Linear Programming

Linear Programming formulation of the same problem can be written as [4],

$$\text{Primal:} \quad \max_{\lambda, h} \lambda$$
$$\text{such that, } B(\lambda \mathbf{1} + h) \leq l + Ph$$

$$\text{Dual:} \quad \min_{\mu \in \mathbb{R}^{\mathcal{X}\mathcal{A}}} \mu^\top l$$
$$\text{such that, } \mu^\top \mathbf{1} = 1, \mu \geq \mathbf{0}, \mu^\top (P - B) = \mathbf{0}$$

- Number of variables and constraints scale with $|\mathcal{X}\mathcal{A}|$
- Approximate Linear Programming(ALP) methods assume the ability to solve an LP with as many constraints as states or access to the stationary distribution from the optimal policy [3].

Approximating Stationary Distribution (Ergodic Cost)

Problem Re-Formulation

Approximating space-action stationary distribution using a parameterized feature matrix such that $\mu = (\mu_0 + \Phi\theta)$. The *dual* problem is reformulated as [2],

$$\min_{\theta} (\mu_0 + \Phi\theta)^\top l$$
$$\text{such that, } (\mu_0 + \Phi\theta)^\top \mathbf{1} = 1, (\mu_0 + \Phi\theta) \geq \mathbf{0}, (\mu_0 + \Phi\theta)^\top (P - B) = \mathbf{1}$$

Above LP can be again reformulated as an ALP names as the expanded efficient large-scale dual ALP,

$$\mu_{\hat{\theta}}^\top l \leq \min \left\{ \mu_\theta^\top l + \frac{1}{\epsilon} V(\theta) : \theta \in \mathrm{R}^d \right\} + O(\epsilon)$$

where, $d \ll |\mathcal{X}\mathcal{A}|$ is the number of features. μ_0 is a known stationary distribution. Φ is a feature matrix of size $(\mathcal{X}\mathcal{A} \times d)$. θ are the parameters and $V(\theta)$ is a violation function for θ.

Recasting to Convex Problem

ALP can be converted into an unconstrained optimization over Θ by adding *constraint violations*.

For a fixed constant $H > 0$ ALP is converted to a convex problem with the cost given as,

$$
\begin{aligned}
c(\theta) = l^\top (\mu_0 + \Phi\theta) \\
&+ H \sum_{(x,a)} \underbrace{\left| \left[\mu_0(x,a) + \Phi_{(x,a),:}\theta \right]_- \right|}_{(\mu_0+\Phi\theta)\geq 0} \\
&+ H \sum_{x'} \underbrace{\left| (P-B)^\top_{:,x'} \Phi\theta \right|}_{(\mu_0+\Phi\theta)^\top (P-B)=1}
\end{aligned}
$$

Gradient Calculation

Calculating the gradients of $c(\theta)$ is still on order of $O(|\mathcal{X}||\mathcal{A}|)$.

An unbiased estimate of gradient can be calculated by sampling (x, a) and x' for T iterations and at round $t = \{1, 2, \ldots, T\}$,

$$\begin{aligned}\nabla c(\theta) \approx g_t(\theta) = & l^\top \Phi - H \frac{\Phi_{(x_t, a_t),:}}{q_1(x_t, a_t)} \mathbb{I}_{\{\mu_0(x_t, a_t) + \Phi_{(x_t, a_t),:}\theta > 0\}} \\ & + H \frac{(P - B)^\top_{:,x'_t} \Phi}{q_2(x'_t)} ((P - B)^\top_{:,x'_t} \Phi\theta)\end{aligned}$$

where q_1 and q_2 are distribution by which (x, a) and x' are sampled respectively.

This estimate is used in the *projected subgradient* algorithm to minimize $c(\theta)$.

Theorem

Consider an expanded efficient large scale dual ALP problem and assume $\tau := \sup\{\tau(\theta) : \theta \in \Theta\} < \infty$ *is finite. Suppose we apply the stochastic subgradient method to the problem. Let* $\epsilon \in (0, 1)$. *Let* $T = \frac{1}{\epsilon^4}$ *be the number of rounds and* $H = \frac{1}{\epsilon}$ *be the constraints multiplier in the subgradient estimate. Let* $\hat{\theta}_T$ *be the output of the method after* T *rounds and let the learning rate be* $\eta_t = \frac{S}{G' \sqrt{T}}$, *where* $G' = \sqrt{d} + H(C_1 + C_2)$. *Then for any* $\delta \in (0, 1)$, *with probability at least* $1 - \delta$,

$$\mu_{\hat{\theta}_T}^\top l \leq \min_{\theta \in \Theta} \left(\mu_\theta^\top l + O\left(\frac{1}{\epsilon} \left(\|[\mu_0 + \Phi\theta]_-\|_1 + \|(P - B)^\top (\mu_0 + \Phi\theta)\|_1 \right) \right) + O(\epsilon) \right)$$

where the constants hidden in the big-O notation are polynomials in S, d, C_1, C_2, *and* $\log(\frac{1}{\delta})$

Approximating Value Function (KL Total Cost)

Kullback-Leibler Total Cost

Kullback-Leibler (KL) loss function is defined as,

$$l(x, P) = q(x) + \sum_{x' \in \mathcal{X}} P(x, x') \log \frac{P(x, x')}{P_0(x, x')}$$

where,

$$\text{arbitrary state cost } q : \mathcal{X} \to [0, Q]$$
$$P \in \mathcal{P}$$
$$\text{fixed } P_0 \in \mathcal{P}$$

Problem Re-Formulation - I

In the optimal setting $(Lh)(x) = h(x)$ [5] where L is the bellman operator. With KL loss function,

$$\arg\min_{P \in \mathcal{P}} \left\{ l(x, P) + \sum_{x' \in \mathcal{X}} P(x, x') h(x') \right\} = \frac{P_0(x, x') e^{-h(x')}}{\sum_{x'} P_0(x, x') e^{-h(x')}}$$

Which gives,

$$(Lh)(x) = q(x) - \log\left(\sum_{x'} P_0(x, x') e^{-h(x')} \right)$$

This considerably simplifies the Bellman optimality equation to:

$$e^{-h(x)} = e^{-q(x)} P_0(x, :) e^{-h(x')}$$

Problem Re-Formulation - II

Taking a family of value functions [1],

$$\mathcal{H} = \{x \mapsto h_w(x) := -\log(\Psi(x,:)w) : w \in \mathcal{W}\}$$

where $\Psi \in \mathbb{R}^{|\mathcal{X}| \times d}$ is a feature matrix and $\mathcal{W} \subset \mathbb{R}^d$ is a bounded set.

The problem can be reformulated in the following constraint problem,

$$\min_{x \in \mathcal{W}} h_w(x_1)$$
$$\text{such that, } e^{-h_w(x)} - e^{-(Lh_w)(x)} = 0, \forall x \in \mathcal{X} 11$$

Here, $e^{-h_w(x)} - e^{-(Lh_w)(x)}$ is the Bellman error $h(x) - (Lh)(x)$ in an exponentiated form.

Recasting to Convex Problem

The constraint optimization problem can be converted to an convex optimization problem by adding constraint violation.

Taking a fixed hyper-parameter $H > 0$ the cost is formulated as,

$$
\begin{aligned}
c(w) = &- \log(\Psi(x_1, :)w) \\
&+ H \sum_{T \in \tau} s(T) \sum_{x \in T} \left| \Psi(x, :)w - e^{-q(x)} P_0(x, :)\Psi w \right|
\end{aligned}
$$

where,

- τ is the set of all *trajectories* starting with state x_1 and ending at an absorbing state z.
- s is the probability distribution over τ.

Gradient Calculation

For large problems it's computationally intractable to sum over all the trajectories τ.

To get an unbiased estimate of the subgradient we sample a trajectory $T \sim s$ (episode of the MDP),

$$\nabla c(w) = r(w) = -\left(\frac{1}{\Psi(x_1,:)w}\right)\Psi(x_1,:) \\ + H\sum_{x\in T}\Big[\operatorname{sign}\left(\Psi(x,:)w - e^{-q(x)}P_0(x,:)\Psi w\right) \\ \left(\Psi(x,:) - e^{-q(x)}P_0(x,:)\Psi\right)\Big]$$

This gradient is used in the *projected subgradient* algorithm to minimize $c(w)$.

Analysis

Theorem

Assume that $\hat{w}$ is ϵ-optimal and choose any $H \geq e^{Q-\log g}$ where $\Psi(x,:)w \geq g \forall x \in \mathcal{X}$. Then, for any $w \in \mathcal{W}$ with $l_w = \min(h_w, Lh_w)$, we have,

$$\begin{aligned} h_{P_{h_{\hat{w}}}}(x_1) - h_{P_{h_w}}(x_1) \leq \epsilon \\ + \|P_{h_{\hat{w}}} - s\|_1 \max_{T\in\tau} \sum_{xinT} |h_{\hat{w}}(x) - Lh_{\hat{w}}(x)| \\ + \sum_{T\in\tau} P_{h_w}(T) \sum_{x\in T} |h_w(x) - Lh_w(x)| \\ + H \sum_{T\in\tau} s(T) \sum_{x\in T} e^{l_w(x)} |h_w(x) - Lh_w(x)| \end{aligned}$$

where, with an abuse of notation, $P_h(T)$ denotes the probability of trajectory T under transition dynamics P_h.

Conclusion

Summary

- Parameterized stationary distribution in the dual problem which hasn't been explored before.
- Parameterized value function without using linear combination basis function which usually is the case.
- Reformulated constraint optimization problems into unconstrained convex optimization.
- Gave unbiased estimates to efficiently calculate subgradient.
- Under weak assumptions, the average stochastic subgradient method produces a parameter competitive to the whole parameter space.

Open Problems

Future Work

- Frame the problem of MDP with absorbing states as stochastic shortest path problem.
- Finding other regulatory/violation function which gives a better bound.
- Control the distribution mismatch between $P_{h_{\hat{w}}}$ and s.

Questions?

References I

Y. Abbasi-Yadkori, P. Bartlett, X. Chen, and A. Malek.
Large-scale markov decision problems with kl control cost and its application to crowdsourcing.
In *International Conference on Machine Learning*, pages 1053–1062, 2015.

Y. Abbasi-Yadkori, P. Bartlett, and A. Malek.
Linear programming for large-scale markov decision problems.
In *Proceedings of the International Conference on Machine Learning*, 2014.

V. V. Desai, V. F. Farias, and C. C. Moallemi.
Approximate dynamic programming via a smoothed linear program.
Operations Research, 60(3):655–674, 2012.

References II

A. S. Manne.
Linear programming and sequential decisions.
Management Science, 6(3):259–267, 1960.

R. S. Sutton and A. G. Barto.
Reinforcement learning: An introduction, 2011.