MS&E 111X/MS&E 211X Suggested Course Project II: Markov Decision Process

Yinyu Ye

October 24, 2021

Markov Decision Processes (MDPs) provide a mathematical framework for modeling sequential decision-making in situations where outcomes are partly random and partly under the control of a decision maker. MDPs are useful for studying a wide range of optimization problems solved via Dynamic Programming (DP), where it was known at least as early as the 1950s (cf. Shapley 1953, Bellman 1957). Modern applications include dynamic planning, reinforcement learning, social networking, and almost all other dynamic/sequential decision game strategy making problems in Mathematical, Physical, Management and Social Sciences.

As talked in class, the MDP problem with m states and total n actions can be formulated as a standard form linear program with m equality constraints and n variables:

$$\begin{array}{lccccc} \min_{\mathbf{x}} & \sum_{j\in\mathcal{A}_1} c_j x_{1j} + & ... & + \sum_{j\in\mathcal{A}_m} c_j x_{mj} & & \\ \text{s.t.} & \sum_{j\in\mathcal{A}_1} (\mathbf{e}_1 - \gamma \mathbf{p}_{j|1}) x_{1j} + & ... & + \sum_{j\in\mathcal{A}_m} (\mathbf{e}_m - \gamma \mathbf{p}_{j|m}) x_{mj} & = & \mathbf{e}, \\ & ... & x_{ij} & ... & \geq & 0, \ \forall i \text{ and } j, \end{array} \tag{1}$$

where $\mathcal{A}_i$ represents the set of all actions available in state i, $\mathbf{p}_{j|i}$ is the state transition probabilities from state i to all states and c_j is the immediate cost when action j is taken, and $0 < \gamma < 1$ is the discount factor. For simplicity, we assume the cost is only dependent on the action but not on the state. Also, $\mathbf{e} \in R^m$ is the vector of ones, and $\mathbf{e}_i$ is the unit vector with 1 at the i-th position and zeros everywhere else. Variable $x_{ij},\ j \in \mathcal{A}_i$, is the state-action frequency or flux, or the expected present value of the number of times in which the process visits state i and takes state-action $j \in \mathcal{A}_i$. Thus, solving the problem entails choosing a state-action frequencies/fluxes that minimize the expected present value sum of total costs. The dual of the

LP is

$$\begin{array}{rlll} \text{maximize}_{\mathbf{y}} & \mathbf{e}^T\mathbf{y} = \sum_{i=1}^m y_i & & \\ \text{subject to} & y_1 - \gamma \mathbf{p}_{j|1}^T \mathbf{y} & \leq & c_j,\ j \in \mathcal{A}_1 \\ & & \vdots & \\ & y_i - \gamma \mathbf{p}_{j|i}^T \mathbf{y} & \leq & c_j,\ j \in \mathcal{A}_i \\ & & \vdots & \\ & y_m - \gamma \mathbf{p}_{j|m}^T \mathbf{y} & \leq & c_j,\ j \in \mathcal{A}_m. \end{array} \tag{2}$$

where y_i represents the cost-to-go value in state i.

Question 1: Explain why that in (1) the basic variables of every basic feasible solution have exactly one variable from each state i. You are not required to mathematically prove this statement and only need to intuitively interpret this statement.

In the VI method, if starting with any vector $\mathbf{y}^0 \geq \mathbf{y}^*$ and assuming $\mathbf{y}^1 \leq \mathbf{y}^0$, then the following entry-wise monotone property holds:

$$\mathbf{y}^* \leq \mathbf{y}^{k+1} \leq \mathbf{y}^k,\ \forall k.$$

This property has been used in a recent paper (see [SWWY17]) on the VI method using samples.

Question 2: Rather than go through all state values in each iteration, we modify the VI method, call it RamdomVI: In the k-th iteration, randomly select a subset of states B^k and do

$$y_i^{k+1} = \min_{j \in \mathcal{A}_i} \{c_j + \gamma \mathbf{p}_{j|i}^T \mathbf{y}^k\},\ \forall i \in B^k. \tag{3}$$

In RandomVI, we only update a subset of state values at random in each iteration.

What can you tell the convergence of the RandomVI method? Does it make a difference with the classical VI method? How does the sample size affect the performance? Simulate computational experiments to verify your claims.

Suppose we build an empirical distribution for each action being selected as the winning action in the final policy: the probability of action j is the past frequency of action j is being selected as the arg min in the previous iterations, e.g., the Bayes update where we start with a uniform distribution $\tilde{p}^0$. Redo the computational experiments by randomly selecting $\mathcal{A}_i$ using the empirical distribution.

Question 3: Here is another modification, called CyclicVI: In the k-th iteration do

- Initialize $\tilde{\mathbf{y}}^k = \mathbf{y}^k$.
- For $i = 1$ to m

$$\tilde{y}_i^k = \min_{j \in \mathcal{A}_i} \{c_j + \gamma \mathbf{p}_{j|i}^T \tilde{\mathbf{y}}^k\} \tag{4}$$

- $\mathbf{y}^{k+1} = \tilde{\mathbf{y}}^k$.

In the CyclicVI method, as soon as a state value is updated, we use it to update the rest of state values.

What can you tell the convergence of the CyclicVI method? Does it make a difference with other VI methods? Use simulated computational experiments to verify your claims.

Question 4: In the CyclicVI method, rather than with the fixed cycle order from 1 to m, we follow a random permutation order, or sample without replacement to update the state values. More precisely, in the kth iteration do

0. Initialize $\tilde{\mathbf{y}}^k = \mathbf{y}^k$ and $B^k = \{1, 2, ..., m\}$
1. – Randomly select $i \in B^k$
   –
$$\tilde{y}_i^k = \min_{j \in \mathcal{A}_i} \{c_j + \gamma \mathbf{p}_{j|i}^T \tilde{\mathbf{y}}^k\} \tag{5}$$
   – remove i from B^k and return to Step 1.
3. $\mathbf{y}^{k+1} = \tilde{\mathbf{y}}^k$.

We call it the randomly permuted CyclicVI or RPCyclicVI in short.

What can you tell the convergence of the RPCyclicVI method? How does it compare with other VI methods? Simulate computational experiments to verify your claims.

Question 5(Optional): How does the structure of the matrix $\mathbf{P}$ affect the convergence? Consider the following choices of $\mathbf{P}$ and discuss what you find:

1. Deterministic MDP. Each state deterministically transitions to another state under a specific action. That is, the vector $p_{j|i}$ is a one-hot vector for all i, j.
2. For a given action j, the transition probabilities to the next state are the same and is not dependent on the current state. That is, the vector $p_{j|i}$ is the same for all i.
3. General Case. The transition probabilities depend on both state and action.

In the experiments, clearly state how you generate the transition probability matrices.

References

[Ber13] Dimitri P Bertsekas. *Abstract dynamic programming.* Athena Scientific, Belmont, MA, 2013.

[HMZ13] Thomas Dueholm Hansen, Peter Bro Miltersen, and Uri Zwick. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *J. ACM*, 60(1):1:1–1:16, February 2013.

[How60] Ronald A. Howard. *Dynamic programming and Markov processes.* The MIT press, Cambridge, MA, 1960.

[LDK95] Michael L Littman, Thomas L Dean, and Leslie Pack Kaelbling. On the complexity of solving markov decision problems. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 394–402. Morgan Kaufmann Publishers Inc., 1995.

[Put14] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming.* John Wiley & Sons, 2014.

[Sch13] Bruno Scherrer. Improved and generalized upper bounds on the complexity of policy iteration. In *Advances in Neural Information Processing Systems*, pages 386–394, 2013.

[SWWY17] Aaron Sidford, Mengdi Wang, Xian Wu, Yinyu Ye. Variance Reduced Value Iteration and Faster Algorithms for Solving Markov Decision Processes. SODA2018 and https://arxiv.org/abs/1710.09988

[Ye11] Yinyu Ye. The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, 36(4):593–603, 2011.