Course Notes On Dynamic Optimization (Fall 2023)
Lecture 8: Online value iteration and optimistic exploration

Instructor: Daniel Russo

Email: djr2174@gsb.columbia.edu

Graduate Instructor: David Cheikhi

Email: d.cheikhi@columbia.edu

These notes are partly based of scribed notes from a previous edition of the class. I have done some follow up light editing, but there may be typos or errors.

1 Problem setup

Throughout these notes we continue to study indefinite horizon problems under the assumption that all policies eventually reach the terminal state.

2 Value iteration

In dynamic programming, the algorithm that itreatively applies the Bellman operator is called value iteration. Here is a version with a fixed iteration count N. In discounted problems, one could set $N \approx \log(1/\epsilon)/(1-\alpha)$ to ensure and ϵ–accurate approximation to J^* is produced. This is overly conservative, however, so in practice you would not use this version of the algorithm, you'd use one that decides when to stop based on looking at the closeness between successive J's.

Algorithm 1 Value iteration

```
Require: $J_0 \in \mathbb{R}^{|\mathcal{X}|}$, MDP $M = (\mathcal{X} \cup \emptyset, \mathcal{U}, g, p, d_0)$, and iteration limit $N$.
for Episode $n = 0, 1, 2, \ldots, N-1$ do
    Reset $J_n^+ \leftarrow J_n$
    for Each state $x \in \mathcal{X}$ do
        Compute $J_n^+(x) \leftarrow \min_{u \in U(x_k^n)} g(x_k^n, u_k^n) + \sum_{x' \in \mathcal{X}} p(x'|x_k^n, u_k^n) J_n(x')$.
    end for
    Update $J_{n+1} \leftarrow J_n^+$
end for
return $J_N$
```

3 Real time dynamic programming

The algorithm requires the following primitives:

- The ability to simulate the state trajectory under a given policy.
- The ability to perform one-step lookahead at a specific state. For instance, we can compute $TJ(x)$.
    - It is worth emphasizing that many problems have large state spaces but sparse transitions. That is, it is easy to loop over all possible successor states from a given state but not to loop over all possible states. RTDP is especially natural in this case.

Algorithm 2 Real-Time Dynamic Programming (RTDP)

```
Require: $J_0 \in \mathbb{R}^{|\mathcal{X}|}$, MDP $M = (\mathcal{X} \cup \emptyset, \mathcal{U}, g, p, d_0)$, and episode limit $N$.
for Episode $n = 0, 1, 2, \ldots,$ do
    Reset $J_n^+ \leftarrow J_n$ and $t \leftarrow 0$
    Sample $x_0^n \sim d_0$
    while $x_k^n \neq \emptyset$ do
        Compute $J_n^+(x) \leftarrow \min_{u \in U(x_k^n)} g(x_k^n, u_k^n) + \sum_{x' \in \mathcal{X}} p(x'|x_k^n, u_k^n) J_n(x')$.
        Choose action $u_k^n \in \arg\min_{u \in U(x_k^n)} g(x_k^n, u_k^n) + \sum_{x' \in \mathcal{X}} p(x'|x_k^n, u_k^n) J_n(x')$
        Apply $u_k^n$ and observe next state $x_{k+1}^n$.
        Update period count: $k \leftarrow k + 1$.
    end while
    Update $J_{n+1} \leftarrow J_n^+$
end for
return $J_N$
```

We denote the policy employed by RTDP by $\mu_n \in G(J_n)$. In other words,

$$\mu_n(x) \in \arg\min_{u \in U(x)} g(x, u) + \sum_{x \in \mathcal{X}} p(x'|x, u) J_n(x')$$

for each x. **A point of possible confusion** is that μ_n is never pre-computed or stored. Instead, we compute it in an real-time fashion by performing one-step lookahead at the states we visit.

4 Illustrating exploration challenges with the 'River Swim' problem

River Swim Problem. We consider the following MDP describing the decision process of one person swimming across the river from land (state 1) to the island (state n) in Figure 1.

Here $\mathcal{X} = \{1, 2, \ldots, n\}$ and $U(x) = \{L, R\}, \forall x \in \mathcal{X}$. The cost function is given by:

$$g(s, L) = 0,$$
$$g(s, R) = \begin{cases} \varepsilon, & \text{if } s \leqslant n - 1 \\ -1, & \text{if } s = n \end{cases}$$

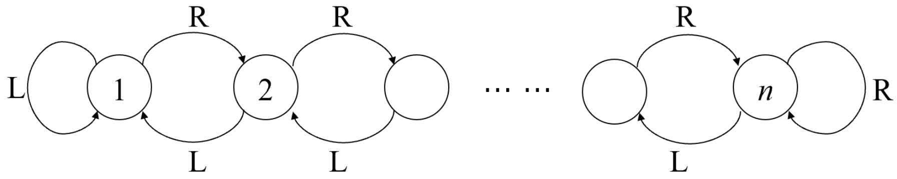

Figure 1: State-action Space of River Swim Problem

And the transition is deterministic, i.e. $p(s-1|s,L)=1, s \geqslant 2; p(s+1|s,R)=1, s \leqslant n-1$. And $p(1|1,L)=p(n|n,R)=1$. It can be easily seen that if ε is small relative to $1-\gamma$, then $\mu^*(x)=R, \forall x$.

The next lemma shows that RTDP fails when applied with a pessimistic initialization.

Lemma 1. *Consider a discounted analogue of the River swim problem (for simplicity). Suppose the initial state is always the leftmost state (i.e. $d_0(1)=1$) and RTDP is applied with initial cost-to-go function $J_0=\vec{0}$. Then, RTDP only visits state 1, always selects action L, and its cost-to-go functions satisfy $J_n=\vec{0}$ for each n.*

Proof. Consider the Bellman update at state 1:

$$TJ_0(1) = \max\{0 + \alpha J_0(1)\, \epsilon + \alpha J_0(2)\} = 0 + J_0(1) = 0.$$

From this calculation $J_0^+(1)=0$, so the Bellman update does not change the cost-to-go estimate. Moreover, $u_0^1 = L$, so the next state is $x_1^0 = 1$. Repeating this observation, we see the system stays in the leftmost state indefinitely and the cost-to-go function never changes. □

One possible resolution to this issue is to enforce that the initial distribution should place strictly positive probability on each initial state. We show that a different, optimistic, initialization, completely resolves the issue.

5 Regret bound under optimism

Right now, my proof requires the following condition to get the tightest bound. Looser bounds do not require this condition.

Definition 1. We say the state transitions are acyclic if, under any policy, no state is visited more than once during an episode.

Here is the key proposition.

Proposition 1. *Suppose transitions are acylcic; If RTDP is applied with $J_0 \preceq J^*$, then,*

$$\sup_{N\in\mathbb{N}} \underbrace{\mathbb{E}\left[\sum_{n=0}^{N-1} \left(J_{\mu_k}(s_0^n) - J^*(x_0^n)\right)\right]}_{\textit{Cummulative Regret}} \leqslant \underbrace{\|J^* - J_0\|_1}_{\textit{Initialization error}} . \tag{1}$$

5.1 First analysis step: preservation of optimism

Lemma 2 (Preservation of Optimism)**.** *Under RTDP, If $J_0 \preceq J^*$, then $J_0 \preceq J_1 \preceq \cdots \preceq J_n \preceq J^*, \forall n$ with probability 1.*

Proof. We prove this result by induction. And we use $n = 0$ as a base of induction. Assume $J_{n-1} \preceq J^*$, then we have:

$$TJ_{n-1} \preceq TJ^* = J^*,$$

where the first inequality follows from the monotonicity of the Bellman operator and the equality follows from the fact that J^* is the fixed point of T. Then we discuss the following two situations with respect to the state $x \in \mathcal{X}$:

- If $x \in \{x_0^n, \cdots, x_{\tau_n}^n\}$, then the value at state x is updated and $J_n(x) = TJ_n(x) \leqslant J^*(x)$ by the above inequality.
- otherwise, the value at state x is not updated and $J_n(x) = J_{n-1}(x) \leqslant J^*(x)$ by the induction hypothesis.

□

5.2 Second analysis step: relating regret to the Bellman gap at visited states

Lemma 3. *Let $J \in \mathbb{R}^{|\mathcal{X}|}$ satisfy $J \preceq J^*$ and the stationary policy $\mu \in G(J)$ be greedy with respect to J. Then*

$$J^\mu(x) - J^*(x) \leqslant \mathbb{E}\left[\sum_{k=0}^{\tau} \left(TJ(x_k) - J(x_k)\right) \mid x_0 = x\right].$$

Proof. Write $P_\mu \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ for the sub-stochastic transition matrix under μ, $P_\mu(x, x') = p(x'|x, \mu(x))$. If $g_\mu \in \mathbb{R}^{|\mathcal{X}|}$, then $T_\mu J = g_\mu + P_\mu J$. We find,

$$\begin{aligned}
J^\mu - J^* &\preceq J^\mu - J && \text{(by optimism)}\\
&= T_\mu J^\mu - J && \text{(since } J_\mu \text{ is a fixed point)}\\
&= \left(T_\mu J^\mu - T_\mu J\right) + \left(T_\mu J - J\right)\\
&= \left(T_\mu J^\mu - T_\mu J\right) + \left(TJ - J\right) && \text{(since } \mu \in G(J))\\
&= \left(g_\mu + P_\mu J^\mu - g_\mu - P_\mu J\right)\\
&= P_\mu\left(J^\mu - J\right) + \left(TJ - J\right)\\
&= \cdots\\
&= \sum_{k=0}^{\infty} P_\mu^k \left(TJ - J\right) && \text{(by iterating the recursion).}
\end{aligned}$$

For any x, $\left(\sum_{k=0}^{\infty} P_\mu^k (TJ - J)\right)(x) = \mathbb{E}\left[\sum_{k=0}^{\tau} \left(TJ(x_k) - J(x_k)\right) \mid x_0 = x\right]$, completing the proof. □

5.3 Completing the proof

When reading this proof, it is helpful to keep in mind that

$$J_0 \preceq J_1 \preceq \cdots \preceq J_n \preceq \cdots \preceq J^*.$$

The main idea in the proof is to relate regret in episode n to the overall reduction in optimism $\sum_{x\in\mathcal{X}}(J_{n+1}(x) - J_n(x))$

Proof.

$$\underbrace{J^{\mu_n}(x_0^n) - J^*(x_0^n)}_{\text{Regret}} \leqslant \mathbb{E}\left[\sum_{k=0}^{\tau_n} (TJ_n(x_k^n) - J_n(x_k^n)) \mid x_0^n\,,\, J_n\right] \tag{2}$$

$$= \underbrace{\mathbb{E}\left[\sum_{k=0}^{\tau_n} (J_{n+1}(x_k^n) - J_n(x_k^n)) \mid x_0^n\,,\, J_n\right]}_{\text{Reduction in optimism}}. \tag{3}$$

Define e to be a vector all 1's, so $\langle e\,,\, J\rangle = \sum_{x\in\mathcal{X}} J(x)$. Then,

$$\sum_{k=0}^{\tau_n} (J_{n+1}(x_k^n) - J_n(x_k^n)) = \langle e\,,\, J_{n+1} - J_n\rangle,$$

using that $J_{n+1}(x) = J_n(x)$ for any $x \notin \{x_0^n, \cdots, x_{\tau_n}^n\}$ and also using for the first (and last) time that state transitions are acyclic). Therefore

$$J^{\mu_n}(x_0^n) - J^*(x_0^n) \leqslant \mathbb{E}\left[\langle e\,,\, J_{n+1} - J_n\rangle \mid x_0^n\,,\, J_n\right].$$

Summing over n and using the law of iterated expectations gives,

$$\mathbb{E}\left[\sum_{n=0}^{N-1} (J^{\mu_n}(x_0^n) - J^*(x_0^n))\right] \leqslant \mathbb{E}\left[\sum_{n=0}^{N-1} \langle e\,,\, J_{n+1} - J_n\rangle\right] = E\left[\langle e\,,\, J_N - J_0\rangle\right].$$

Since $J_0 \preceq J_N \preceq J^*$, we upper bound the final term with probability 1 as

$$\langle e\,,\, J_N - J_0\rangle \leqslant \langle e\,,\, J^* - J_0\rangle = \|J^* - J_0\|_1.$$

□