Reinforcement Learning Lecture Notes:
Policy Evaluation from Data

Mohammad Sadegh Talebi
University of Copenhagen
sadegh.talebi@di.ku.dk

Initial Draft: February 11, 2024
This Version: February 19, 2024

1 Introduction

In this note, we study policy evaluation and off-policy evaluation in infinite-horizon discounted MDPs. The main of these methods is to evaluate the quality of a target policy (a policy in question) via the data collected from the MDP under a behavior policy (which might be different from the target policy).

Notations. We introduce notations that will be used throughout. Given a set A, $\Delta(A)$ denotes the simplex of probability distributions over A. $\mathbb{I}\{A\}$ denotes the indicator function of event A, namely it equals 1 if A occurs, else 0. $\mathbb{N}$ and $\mathbb{R}$ denotes the set of naturals and reals, respectively. Unless specified, $\|\cdot\|$ is used to denote a generic norm. We write $X \sim p$ to indicate that X is drawn randomly from distribution p. Finally, given a set A, $|A|$ denotes the cardinality of A, and a function $f : A \to \mathbb{R}^{|A|}$ could interchangeably be viewed as a $|A|$-dimensional vector.

2 Problem Formulation

Consider an infinite-horizon discounted MDP $M = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote state-space and action-space, respectively, where for simplicity (and without loss of generality) we assume that the set of available actions is the same at all states. $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{A})$ and $R : \mathcal{S} \times \mathcal{A} \to \Delta(\mathbb{R})$ denote the transition and reward functions, respectively. Finally, $\gamma \in (0, 1)$ is the discount factor. We assume that the reward function R satisfies the bounded reward assumptions.

Given a (fixed) policy $\pi \in \Pi^{\mathrm{SR}}$, the interaction with M goes at follows. At $t = 1$, the MDP is in some initial state $s_1 \in \mathcal{S}$ (e.g., chosen by Nature). At each time $t \geq 1$, the agent observes the current state $s_t \in \mathcal{S}$ and chooses $a_t \sim \pi(\cdot|s_t)$. Upon executing a_t, M samples a next-state $s_{t+1} \sim P(\cdot|s_t, a_t)$ and a reward $r_t \sim R(s_t, a_t)$. The agent receives r_t by the end of the current slot. And this process continues.

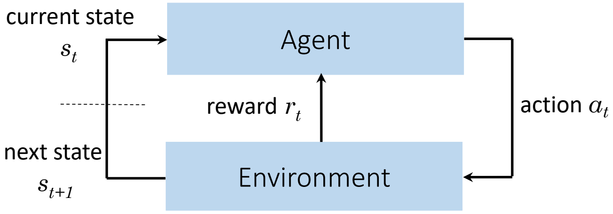

Figure 1: The interaction between the agent and the environment (MDP)

The value of π, denoted by V^π, is defined as $V^\pi(s) = \mathbb{E}\big[\sum_{t=1}^\infty r_t \big| s_1 = s\big]$. Further, we recall that following the Bellman equation (for π), we have: $V^\pi = r^\pi + \gamma P^\pi V^\pi$, which leads to

$$V^\pi = (I - \gamma P^\pi)^{-1} r^\pi, \tag{1}$$

where P^π and r^π denote the transition matrix and reward vector (function) associated to the MRP induced by π on M:

$$\forall s, s' \in \mathcal{S}: \qquad P^\pi_{s,s'} = \sum_{a\in\mathcal{A}} \pi(a|s)P(s'|s,a), \quad r^\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s)R(s,a) \tag{2}$$

2.1 Policy Evaluation from Data

Policy evaluation (PE) is the task of computing the value V^π of a given policy π. When the MDP M is known (i.e., P and R are specified), one can compute V^π using (1). However, there are situations in practical scenarios where M is not known beyond its state-action space (or partially known), but we can generate trajectories from M. In this respect, M can be viewed as a black-box which can be queried with an action to return a next-state and a reward value, following the aforementioned model.

The task of **policy evaluation from data** is defined as that of estimating the value of a given π through data generated as above. More precisely, it is assumed that a dataset $\mathcal{D} = \big\{(s_t, a_t, r_t), 1 \le t \le n\big\}$ is given where

$$a_t \sim \pi(\cdot|s_t), \quad r_t \sim R(s_t, a_t), \quad s_{t+1} \sim P(\cdot|s_t, a_t)$$

and the goal is to derive a point estimate $\widehat{V}^\pi$ of V^π, potentially together with some confidence interval acting as certificates of the derived estimate. The confidence interval could say how far $\widehat{V}^\pi$ is from V^π with probability $1-\delta$ for some $\delta \in (0,1)$. The dataset could be formed by multiple trajectories (i.e., under multiple independent interactions). However, for simplicity, we assume that $\mathcal{D}$ contains a single trajectory.

In some decision making scenarios, it might be possible to derive parameters of an MDP M as the phenomena governing them could be fully characterized. However, there are a plethora of situations where the underlying parameters, P and R, are unknown or imperfectly known, yet a large amount of data from interaction with MDP is available. In such situations, the above mentioned problem makes perfect sense.

2.2 Off-Policy Evaluation and Optimization

We might be interested in evaluating another policy than that used to generate the data. In **off-policy evaluation (OPE)**, the goal is to estimate the value of a **target policy** π (a.k.a. evaluation policy), using data collected under a **behavior policy** π_b (a.k.a. logging policy). Arguably, OPE admits a broader range of applications than PE since historical data collected from an MDP under a behavior policy could be used for multiple purposes (i.e., multiple target policies). Further, directly executing a target policy might be expensive or even risky before its value is estimated. Whereas data could be collected under a cheaper (and safer) behavior policy, or it might already exist. Finally, note that we might have collected data under multiple behavior policies.

For example, consider an RL scenario where a company sells a product according to fixed policy π_1. This situation can be modeled as an MDP. Even though the company may not know the underlying MDP perfectly, PE could be used to derive accurate estimates of the value of π_1, representing the revenue under π_1, under mild assumptions provided that enough data under π_1 is collected. Now suppose that the company is interested in adopting a new policy π_2 with the hope of generating more revenue (i.e., $V^{\pi_2} \ge V^{\pi_1}$ or practically $V^{\pi_2} \ge V^{\pi_1} + \Delta$ for some margin Δ). It might not render rational to try π_2 directly as it may result in a lower revenue. Instead, under some assumptions that will be formalized later, one may use data collected under π_1 to derive estimates of V^{π_2}, using algorithms for OPE. In some cases, we might be content just by knowing $\widehat{V}^{\pi_2} \ge \widehat{V}^{\pi_1} + \Delta$, thereby trusting our estimates. However, the company board may grant switching to π_2 only if *lower confidence bound* V^{π_2} exceeds an *upper confidence bound* on V^{π_1} (and potentially with some margin as input) with high probability.

Another related task is **off-policy optimization (OPO)** where the goal is to find an optimal policy (or an ε-optimal one) via data collected under a behavior policy.

In most literature on RL, OPE and OPO are generally referred as **off-policy learning**, where "off" is used to signify that learning takes place without collecting samples from the policy in question, be it a specified target policy or a sought near-optimal policy. This stands in contrast to **on-policy learning** where learning relies on data collected using the policy of interest.

Returning to the example above, OPO could be used to find a near-optimal policy, whose revenue is nearly-maximal, only relying on data collected under π_1. As we will see later, there exist algorithms capable of doing so provided that π_1 is *exploratory enough* – i.e., performs enough explorations – in some mathematical sense.

3 Algorithms for Policy Evaluation

We present two algorithms for the problem of policy evaluation from data in finite discounted MDPs.

3.1 A Model-based Approach

If the transition function P and the reward function R were known, one would compute P^π and r^π associated to the input policy π and then compute V^π using (1). Hence, a natural approach is to estimate P and R from the dataset $\mathcal{D}$, and assert the so-called **certainty equivalence principle**, i.e., proceed as if these estimates equal their true values. In view of (1), it would suffice to estimate P^π and r^π.

In order to define empirical estimates for P^π and r^π, we introduce some necessary notations. Given $\mathcal{D}$, define for all $s, s' \in \mathcal{S}$,

$$N(s,s') = \sum_{t=1}^{n-1} \mathbb{I}\{s_t = s, s_{t+1} = s'\} \quad \text{and} \quad N(s) = \sum_{s' \in \mathcal{S}} N(s,s')\,.$$

In words, $N(s)$ denotes the visit count to state s in $\mathcal{D}$, whereas $N(s,s')$ captures the visit count to state s followed by a visit to state s'. We consider **smoothed estimates** for P^π and r^π defined as follows:

$$\forall s, s' \in \mathcal{S}: \qquad \widehat{P}^\pi_{s,s'} = \frac{N(s,s') + \alpha}{N(s) + \alpha S}, \quad \widehat{r}^\pi(s) = \frac{\alpha + \sum_{t=1}^{n-1} r_t \mathbb{I}\{s_t = s\}}{\alpha + N(s)}\,.$$

Here, $\alpha \geq 0$ is an arbitrary choice controlling the level of smoothing. The case of $\alpha = 0$ (i.e., no smoothing) corresponds to **Maximum Likelihood estimator**, which results in unbiased estimates of P^π and r^π.

The case of $\alpha = 1/S$ corresponds to **Laplace smoothing**, which yields biased estimates, but the associated biases vanish as $N(s)$ increases. To see the role of smoothing, consider $\alpha = 0$. Now, if some state s is not visited in $\mathcal{D}$ – hence $N(s) = 0$ – the estimates above are not well-defined. In contrast, with $\alpha > 0$, the estimates above are always well-defined.

Having defined $\widehat{P}^\pi$ and $\widehat{r}^\pi$, we derive the following estimate of V^π, in the spirit of certainty equivalence principle applied to (1):

$$\widehat{V}^\pi = (I - \gamma \widehat{P}^\pi)^{-1} \widehat{r}^\pi \tag{3}$$

Note that each choice of α leads to a different $\widehat{V}^\pi$.

The method presented above follows a **model-based** approach since it maintains an approximate model of the MDP (or the MRP associated to π) and then computes V^π for that. In the case of $\alpha = 0$, the estimate $\widehat{V}^\pi$ may be called a **plug-in estimate**.

We shall call this method Model-Based PE (`MB-PE`). Its pseudo-code is presented as Algorithm 1.

Algorithm 1 Model-Based PE (MB-PE)

```
Input: $\mathcal{D}, \alpha$
For all $s, s' \in \mathcal{S}$, compute
    $N(s,s') = \sum_{t=1}^{n-1} \mathbb{I}\{s_t = s, s_{t+1} = s'\}, \qquad N(s) = \sum_{s' \in \mathcal{S}} N(s,s')$
For all $s, s' \in \mathcal{S}$, compute
    $\widehat{P}^{\pi}_{s,s'} = \frac{N(s,s') + \alpha}{N(s) + \alpha S}, \qquad \widehat{r}^{\pi}(s) = \frac{\alpha + \sum_{t=1}^{n-1} r_t \mathbb{I}\{s_t = s\}}{\alpha + N(s)}$
Compute $\widehat{V}^{\pi} = (I - \gamma \widehat{P}^{\pi})^{-1} \widehat{r}^{\pi}$
Output: $\widehat{V}^{\pi}$
```

Convergence Guarantee. By the strong law of large numbers, $\widehat{P}^{\pi}_{s,s'}$ (resp. $\widehat{r}^{\pi}(s)$) converges to $P^{\pi}_{s,s'}$ (resp. r^{π}) as $N(s)$ tends to infinity, almost surely. Hence, the following result follows:

Theorem 1 *If all states are visited infinitely often under π, then $\widehat{V}^{\pi}$ converges to V^{π} almost surely:*

$$\mathbb{P}\Big(\lim_{n\to\infty} \widehat{V}^{\pi} = V^{\pi}\Big) = 1$$

For the above theorem to hold, all states must be visited infinitely often under π, i.e., $N(s) \to_{n\to\infty} \infty$ for all $s \in \mathcal{S}$. In other words, if π is *exploratory enough*, $\widehat{V}^{\pi}$ converges to V^{π} almost surely. Theorem 1 presents a strong guarantee on V^{π}, but is valid only asymptotically (in terms of the size n of $\mathcal{D}$). More interesting would non-asymptotic convergence guarantees, albeit weaker than Theorem 1. Specifically, one could derive non-asymptotic results using concentration inequalities (e.g., Hoeffding's or Bernstein's inequalities) which would provide error certificates. For instance, such results could be used to determine n, such that $\|V^{\pi} - \widehat{V}^{\pi}\|_{\infty} \le \varepsilon$ with probability higher than $1 - \delta$ for some pre-specified $\delta \in (0, 1)$.

MB-PE converges fast in practice. Further, it follows a simple yet intuitive design. However, it has some drawbacks. First, it yields value estimates with a large variance in practice, which is undesirable. It space complexity (i.e., the size of working memory needed) is $O(S^2)$ as it needs to maintain $S^2 + S$ elements of the estimated MRP. However, note that the size of the quantity of interest, V^{π}, is S. In terms of computational complexity, it involves a matrix inversion which costs $O(S^3)$, which might be large in MDPs with large state-spaces. Finally, MB-PE may not be easily converted to an incremental procedure.

3.2 A Model-free Approach: Temporal Difference Learning

We present a second algorithm for policy evaluation called **Temporal Difference** (TD) learning. TD learning was popularized and extended by Richard Sutton in 1988 [1]. However, the approach is older and the earliest reported use dates back to 1959, where Arthur Samuel, an AI pioneer, used it to solve games [2].

To introduce TD, let us suppose that some estimate $\widehat{V}$ of V^{π} is available and consider $(s_t, a_t, r_t, s_{t+1}) \in \mathcal{D}$. Hence, $\widehat{V}(s_t)$ is an estimate for $V^{\pi}(s_t)$. Now, consider $V'(s_t) = r_t + \gamma\widehat{V}(s_{t+1})$, built using $\widehat{V}$ and (s_t, a_t, r_t, s_{t+1}). We have

$$\mathbb{E}\Big[V'(s_t)\Big|s_t, \widehat{V}\Big] = \mathbb{E}\Big[r_t + \gamma\widehat{V}(s_{t+1})\Big|s_t, \widehat{V}\Big] = \mathbb{E}_{a\sim\pi(\cdot|s_t)}\Big[R(s_t, a) + \gamma \sum_{s'} P(s'|s_t, a)\widehat{V}(s')\Big|s_t, \widehat{V}\Big] .$$

Here we condition on $\widehat{V}$ as it might be random (as it depends on earlier data samples). Hence, $V'(s_t) = r_t + \gamma\widehat{V}(s_{t+1})$ serves as *another estimate* for $V^{\pi}(s_t)$. In other words, using (s_t, a_t, r_t, s_{t+1}), we could refine the estimate value of state s_t, $\widehat{V}(s_t)$, using another estimate $V'(s_t)$. This is a form of *bootstrapping*.

Which of the estimates, $V'(s_t)$ and $\widehat{V}(s_t)$, should we trust? Ideally, we would like to have an estimate $\widehat{V}$ so that the two values nearly coincide, namely,

$$\widehat{V}(s_t) \approx r_t + \gamma\widehat{V}(s_{t+1})$$

As $V'(s_t) = r_t + \gamma\widehat{V}(s_{t+1})$ is constructed using a further data sample, in view of Bellman's equation $r_t + \gamma\widehat{V}(s_{t+1})$ could be interpreted as a target estimate for $V^\pi(s_t)$. This leads us to defining the notion of *temporal difference* error:

$$\delta_t = r_t + \gamma\widehat{V}(s_{t+1}) - \widehat{V}(s_t)\,.$$

The error δ_t serves as a measure of estimation error. We may thus update $\widehat{V}(s_t)$ to reduce the error δ_t:

$$\underbrace{\widehat{V}(s_t)}_{\text{new value}} \longleftarrow \underbrace{\widehat{V}(s_t)}_{\text{old value}} + \alpha_t \underbrace{\Big(r_t + \gamma\widehat{V}(s_{t+1}) - \widehat{V}(s_t)\Big)}_{\text{estimation error}}$$

where α_t is a step-size parameter (a.k.a. learning rate), which is chosen sufficiently small to guarantee the convergence, as we shall explain momentarily.

The update equation above is the core of TD learning, which is a form of bootstrapping. Algorithm 2 presents the pseudo-code of TD.

Algorithm 2 Temporal Difference Learning (TD)

```
input: $\mathcal{D}$, $(\alpha_t)_{t\ge 1}$
Initialize: Select $V_1$ arbitrarily
for $t = 1, \dots, n-1$ do
    Update:
        $V_{t+1}(s) = \begin{cases} V_t(s) + \alpha_t\Big(r_t + \gamma V_t(s_{t+1}) - V_t(s)\Big) & s = s_t \\ V_t(s) & \text{else.} \end{cases}$
end for
Output: $V_n$
```

TD is **model-free** method in the sense that it does not require a model of the MDP (or the MRP associated to π), but rather direct maintains a value function estimate.

Remark 1 *The update equation of TD resembles that of gradient methods for optimization (more precisely, stochastic gradient descent). So this question naturally arises that if TD is a gradient algorithm (and if so, which function it optimizes)? As it turns out, it can be shown that TD is not a gradient method for any objective function.*

Convergence of TD. In fact, TD is a **stochastic approximation** algorithm [3] and as such, it inherits its convergence guarantee from stochastic approximation methods. To have theoretical convergence guarantees of TD, the sequence of learning rates $(\alpha_t)_{t\ge 1}$ must satisfy the **Robbins-Monro conditions** [3]:

$$\alpha_t > 0, \qquad \sum_{t=1}^{\infty} \alpha_t = \infty, \qquad \sum_{t=1}^{\infty} \alpha_t^2 < \infty$$

In other words, it is a positive sequence that is *square-summable-but-not-summable.* Examples include: (i) $\alpha_t = \frac{1}{t+1}$; (ii) $\alpha_t = \frac{2}{\sqrt{t}\log(t+1)}$; and (iii) $\alpha_t = \frac{c}{t^a+2}$ for $a \in (\frac{1}{2}, 1]$ and $c > 0$.

Further, note that one could choose $\alpha_{N_t(s)}$ instead of α_t, where $N_t(s)$ denotes the visit count of state s up to time t. This choice often leads to substantially faster convergence in practice.

The following theorem establishes asymptotic convergence of $(V_t)_{t\ge 1}$ produced by TD:

Theorem 2 *If all states are visited infinitely often under π and $(\alpha_t)_{t\ge 1}$ satisfies the Robbins-Monro conditions, then V_t converges to the V^π almost surely:*

$$\mathbb{P}\left(\lim_{t\to\infty} V_t = V^\pi\right) = 1$$

Similar to Theorem 1, to guarantee convergence to V^π, all states must be visited infinitely often under π, i.e., $N(s) \to_{n\to\infty} \infty$ for all $s \in \mathcal{S}$. In other words, if π is *exploratory enough*. Theorem 2 presents a strong guarantee on V^π, but is valid only asymptotically (in terms of the size n of $\mathcal{D}$). More interesting would non-asymptotic convergence guarantees, albeit weaker than Theorem 2. Specifically, one could derive non-asymptotic results using concentration inequalities (e.g., Hoeffding's or Bernstein's inequalities) which would provide error certificates. For instance, such results could be used to determine n, such that $\|V^\pi - \widehat{V}^\pi\|_\infty \le \varepsilon$ with probability higher than $1-\delta$ for some pre-specified $\delta \in (0,1)$.

We conclude with some remarks. `TD` can be slower than `MB-PE` in practice. However, it often produces value function estimates with a lower variance. A positive feature of `TD` is that it can be incremental (unlike `MB-PE`), i.e., the output estimate could be incrementally refined if new piece of data is available, without the need to rerun the algorithm. Computational complexity (per-step) is $O(1)$. It has a space complexity of S, which is much lower than that of `MB-PE`.

Acknowledgement. I would like to thank Chenxiao Ma for their comments on earlier drafts.

References

[1] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3:9–44, 1988.

[2] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959.

[3] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.