Master MVA: Reinforcement Learning **Lecture: 5**

Approximate Dynamic Programming

Lecturer: *Alessandro Lazaric* *http://researchers.lille.inria.fr/∼lazaric/Webpage/Teaching.html*

Objectives of the lecture

1. **Understand:** The role of approximation in dynamic programming algorithms.
2. **Use:** Approximate value iteration, approximate policy iteration.

1 Dynamic Programming with Approximation

The dynamic programming algorithms introduced in Lecture 2 allow to compute the optimal value function V^* and the optimal policy π^* using value or policy iteration schemes. In practice, this is often not possible and the optimal solutions can only be approximated. In particular, the two main sources of approximation are:

- *Representation approximation.* So far we considered the finite MDP case when $|X| = N$, which implies that $V \in \mathbb{R}^N$. If N is large (or X is continuous), we need to store and update a large number of parameters to represent the functions that we want to learn. In some applications, this is not possible and we need to define an *approximation space* which can represent functions on X in a **compact** way. This restricts the set of functions that we can actually learn and it introduces an *approximation error* (or bias).
- *Sampling approximation.* Dynamic programming algorithms assume that the dynamics and reward are perfectly known. In Lecture 3 we studied how this assumption can be relaxed using reinforcement learning algorithms. Nonetheless, these algorithms are guaranteed to converge to the exact value function only asymptotically. When only a finite number of samples is available, these methods have an approximation due to the non exact estimation of the value function (or Q-function). This second source of approximation is referred to as *estimation error* (or variance).

In this lecture we study how approximations influence the performance of value and policy iteration algorithms and the guarantees that we can still provide.

Notice: in this lecture we only focus on the setting of infinite horizon with discount γ.

2 Performance Loss and Value Function Approximation

We want to study the impact of an approximation of V^* in terms of the performance of the greedy policy. Let V be an approximation of V^*, the greedy policy w.r.t. V is defined as

$$\pi(x) \in \arg\max_{a \in A} \sum_y p(y|x,a)\big[r(x,a,y) + \gamma V(y)\big].$$

We want to provide a relationship between the approximation of V and the performance of π (in terms of its value function V^π).

Proposition 1. We consider the discounted infinite horizon setting. Let $V \in \mathbb{R}^N$ be any function on X and π its corresponding greedy policy, then

$$\|V^* - V^\pi\|_\infty \leq \frac{2\gamma}{1-\gamma}\|V - V^*\|_\infty.$$

Furthermore, there exists $\epsilon > 0$ such that if $\|V - V^*\|_\infty \leq \epsilon$, then π is optimal.

Proof. We have the following sequence of inequalities

$$\begin{aligned}
\|V^* - V^\pi\|_\infty &\overset{(a)}{\leq} \|\mathcal{T}V^* - \mathcal{T}^\pi V\|_\infty + \|\mathcal{T}^\pi V - \mathcal{T}^\pi V^\pi\|_\infty \\
&\overset{(b)}{\leq} \|\mathcal{T}V^* - \mathcal{T}V\|_\infty + \gamma\|V - V^\pi\|_\infty \\
&\overset{(c)}{\leq} \gamma\|V^* - V\|_\infty + \gamma(\|V - V^*\|_\infty + \|V^* - V^\pi\|_\infty) \\
&\overset{(d)}{\leq} \frac{2\gamma}{1-\gamma}\|V^* - V\|_\infty.
\end{aligned}$$

(a) By definition of Bellman operators, V^* and V are fixed points of $\mathcal{T}$ and T^π, i.e., $V^* = \mathcal{T}V^*$ and $V^\pi = \mathcal{T}^\pi V^\pi$. Then application of triangle inequality.

(b) By definition of greedy policy $\mathcal{T}V = \mathcal{T}^\pi V$. Furthermore, $\mathcal{T}^\pi$ is a contraction in L_∞-norm.

(c) By contraction of $\mathcal{T}$ and triangle inequality.

(d) Reordering.

Let $\delta = \min_\pi \|V^\pi - V^*\|_\infty$ be the minimum gap between the optimal value function and the value of any other (non-optimal) policy. Since the number of state and actions is finite, $\delta > 0$. Let $\epsilon > 0$ be such that

$$\frac{2\gamma}{1-\gamma}\epsilon < \delta,$$

then if $\|V - V^*\|_\infty \leq \epsilon$, it follows that $\|V^* - V^\pi\|_\infty < \delta$, which implies that π is optimal. □

While in the this section we analyzed the quality of the greedy policy for any approximation V, in the next sections we study what are the possible ways to actually compute an approximation.

3 Bellman Residual Minimization

Let $\mathcal{F}$ be a function space equipped with a norm $\|\cdot\|$. In the case of $|X| = N$, $\mathcal{F}$ is a vector space in $\mathbb{R}^N$. Let $B(V)$ the (norm of the) Bellman residual of a function V, defined as $B(V) = ||V - \mathcal{T}V||$. We notice that the optimal value function V^* is the fixed point of the Bellman operator $\mathcal{T}V = V$ and thus it is the only function with zero Bellman residual $B(V^*) = 0$. Thus, we deduce that a function with small Bellman residual is likely to be a good approximation of V. In particular, we study the property of the function $V \in \mathcal{F}$ which minimizes the Bellman residual, i.e.,

$$\inf_{V \in \mathcal{F}} \|\mathcal{T}V - V\|.$$

3.1 Approximation Error

We consider the L_∞-norm and we want to relate the approximation error $\|V^* - V\|_\infty$ and the performance error $\|V^* - V^\pi\|_\infty$ (with π the greedy policy w.r.t. V) as a function of the Bellman residual $B(V) = \|\mathcal{T}V - V\|_\infty$.

Proposition 2. *[Williams et Baird, 1993]* Let $V \in \mathbb{R}^n$ be any function, then,

1. We have
$$\|V^* - V\|_\infty \leq \frac{1}{1-\gamma}\|\mathcal{T}V - V\|_\infty. \tag{1}$$
2. Let π be the greedy policy w.r.t. V, then
$$\|V^* - V^\pi\|_\infty \leq \frac{2}{1-\gamma}\|\mathcal{T}V - V\|_\infty.$$
3. Let assume that there exists a minimizer of the Bellman residual in $\mathcal{F}$, i.e., $V_{BR} = \arg\min_{V \in \mathcal{F}} \|\mathcal{T}V - V\|_\infty$. Then
$$\|\mathcal{T}V_{BR} - V_{BR}\|_\infty \leq (1+\gamma) \inf_{V \in \mathcal{F}} \|V^* - V\|_\infty. \tag{2}$$

Finally, combining 2 and 3, and defining π_{BR} as the greedy policy w.r.t. V_{BR}, we have

$$\|V^* - V^{\pi_{BR}}\|_\infty \leq \frac{2(1+\gamma)}{1-\gamma} \inf_{V \in \mathcal{F}} \|V^* - V\|_\infty.$$

Proof. Statement 1. We have the following sequence of inequalities

$$\begin{aligned}
\|V^* - V\|_\infty &\overset{(a)}{\leq} \|V^* - \mathcal{T}V\|_\infty + \|\mathcal{T}V - V\|_\infty \\
&\overset{(b)}{\leq} \gamma\|V^* - V\|_\infty + \|\mathcal{T}V - V\|_\infty \\
&\overset{(c)}{\leq} \frac{1}{1-\gamma}\|\mathcal{T}V - V\|_\infty
\end{aligned}$$

(a) Triangle inequality.

(b) Contraction of $\mathcal{T}$.

(c) Reordering.

Statement 2. By triangle inequality we have $\|V^* - V^\pi\|_\infty \leq \|V^* - V\|_\infty + \|V - V^\pi\|_\infty$. Then we focus on $\|V - V^\pi\|_\infty$. We have

$$\begin{aligned}\|V - V^\pi\|_\infty \leq & \|V - \mathcal{T}V\|_\infty + \|\mathcal{T}V - V^\pi\|_\infty \\ \leq & \|\mathcal{T}V - V\|_\infty + \gamma\|V - V^\pi\|_\infty \\ \leq & \frac{1}{1-\gamma}\|\mathcal{T}V - V\|_\infty.\end{aligned}$$

Then using statement 1, we obtain

$$\|V^* - V^\pi\|_\infty \leq \frac{2}{1-\gamma}\|\mathcal{T}V - V\|_\infty.$$

Statement 3. By the contraction property we have

$$\begin{aligned}\|\mathcal{T}V - V\|_\infty &\leq \|\mathcal{T}V - V^*\|_\infty + \|V^* - V\|_\infty \\ &\leq (1+\gamma)\|V^* - V\|_\infty.\end{aligned}$$

Thus the minimizer of the Bellman residual satisfies

$$\begin{aligned}\|\mathcal{T}V_{BR} - V_{BR}\|_\infty &= \inf_{V \in \mathcal{F}} \|\mathcal{T}V - V\|_\infty \\ &\leq (1+\gamma) \inf_{V \in \mathcal{F}} \|V^* - V\|_\infty\end{aligned}$$

□

3.2 Implementation

There are a number of issues in implementing the Bellman residual minimization. Let $\mathcal{F} = \{f_\alpha\}$ be a set of functions parameterized by a parameter α, then

- Minimizing over an L_∞-norm is computationally very hard, since it implies that the Bellman residual should be minimal over all the states.
- Even if we move to a weighted $L_{2,\mu}$-norm, with μ an arbitrary distribution over X, the objective function $\alpha \mapsto B(\alpha) = \|\mathcal{T}V_\alpha - V_\alpha\|_{2,\mu}^2$ is not convex.

Thus, we move to using a gradient descent method, which is guaranteed to converge to a local minimum. In particular, we update the parameter α using

$$\alpha \leftarrow \alpha - \eta \nabla B(\alpha)$$

The problem is that the gradient might not have a known functional form. So it needs to be estimated as

1. We draw n states at random from the state distribution μ, $X_i \sim \mu$,
2. We define the empirical Bellman residual for the current vector α as

$$\hat{B}(\alpha) = \frac{1}{n}\sum_{i=1}^{n} \left[\mathcal{T}V_\alpha(X_i) - V_\alpha(X_i)\right]^2$$

and perform a gradient descent on the sub-gradient [1]

$$\nabla_\alpha \hat{B}(\alpha) = \frac{2}{n}\sum_{i=1}^{n}[\mathcal{T}V_\alpha - V_\alpha](X_i)(\gamma P^{\pi_\alpha} - I)\nabla V_\alpha(X_i),$$

where π_α is the greedy policy w.r.t. V_α.

Again, in the general case where the dynamics (P) is unknown, the computation of $\mathcal{T}V_\alpha(X_i)$ and $P^{\pi_\alpha}V_\alpha(X_i)$ might not be simple.

4 Approximate Value Iteration

We recall that the definition of the value iteration algorithm comes from the fact that the optimal value function V^* is the only fixed point of the optimal Bellman operator $\mathcal{T}$. Thus, V^* can be applied by repeatedly applying the operator $\mathcal{T}$ to any initial function V_0, obtaining the value iteration algorithm

$$V_{k+1} = \mathcal{T}V_k.$$

From the contraction property of $\mathcal{T}$ we have that $\|V^* - V_{k+1}\|_\infty \leq \gamma\|V^* - V_k\|_\infty$, which implies that $V_k \to V^*$.

When some form of approximation is introduced, it means that we cannot compute or store correctly the function $\mathcal{T}V_k$ and an approximation error is made.

Algorithm Definition 1. In general, the *approximate value iteration* (AVI) algorithm is defined as

$$V_{k+1} = \mathcal{A}\mathcal{T}V_k,$$

where $\mathcal{A}$ is a generic **approximation operator**.

A standard case for $\mathcal{A}$ is that we constrain the functions to belong to a space $\mathcal{F}$. Then, $\mathcal{A}$ is usually the projection operator of the target function (in this case $\mathcal{T}V_k$) onto the space $\mathcal{F}$. More formally, we have that if $\mathcal{A}$ is the projection operator in some norm $\|\cdot\|$, then

$$V_{k+1} = \arg\inf_{V\in\mathcal{F}} \|\mathcal{T}V_k - V\|. \tag{3}$$

Proposition 3. Let $\mathcal{A}$ be a projection is L_∞-norm, denoted by Π_∞, then $\mathcal{A}$ is a non-expansion and the joint operator $\mathcal{A}\mathcal{T}$ is still a contraction, which guarantees the existence of a unique fixed point $\tilde{V} = \mathcal{A}\mathcal{T}\tilde{V}$ and thus the convergence of AVI.

Notice that for computational reasons, it may be preferable to consider the standard projection operator in a $L_{2,\mu}$-norm, but then $\Pi_{2,\mu}\mathcal{T}$ is not a contraction and AVI is not guaranteed to converge anymore.

4.1 Approximation Error

Proposition 4. *[Bertsekas & Tsitsiklis, 1996]* Let V^K be the function returned by AVI after K iterations

[1] The expression of the gradient follows from the definition of $\mathcal{T}$.

and π_K its corresponding greedy policy. Then the performance error is bounded as

$$\|V^* - V^{\pi_K}\|_\infty \leq \frac{2\gamma}{(1-\gamma)^2} \max_{0\leq k<K} \|\mathcal{T}V_k - \mathcal{A}\mathcal{T}V_k\|_\infty + \frac{2\gamma^{K+1}}{1-\gamma}\|V^* - V_0\|_\infty.$$

Proof. Let $\varepsilon = \max_{0\leq k<K} \|\mathcal{T}V_k - \mathcal{A}\mathcal{T}V_k\|_\infty$. This is the largest approximation error done over the iterations. For any $0 \leq k < K$ we have

$$\begin{aligned} \|V^* - V_{k+1}\|_\infty &\leq \|\mathcal{T}V^* - \mathcal{T}V_k\|_\infty + \|\mathcal{T}V_k - V_{k+1}\|_\infty \\ &\leq \gamma\|V^* - V_k\|_\infty + \varepsilon, \end{aligned}$$

then

$$\begin{aligned} \|V^* - V_K\|_\infty &\leq (1 + \gamma + \cdots + \gamma^{K-1})\varepsilon + \gamma^K\|V^* - V_0\|_\infty \\ &\leq \frac{1}{1-\gamma}\varepsilon + \gamma^K\|V^* - V_0\|_\infty \end{aligned}$$

Since from Proposition 1 we have that $\|V^* - V^{\pi_K}\|_\infty \leq \frac{2\gamma}{1-\gamma}\|V^* - V_K\|_\infty$, then we obtain

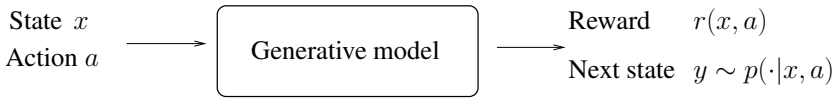
$$\|V^* - V^{\pi_K}\|_\infty \leq \frac{2\gamma}{(1-\gamma)^2}\varepsilon + \frac{2\gamma^{K+1}}{1-\gamma}\|V^* - V_0\|_\infty.$$

□

4.2 Implementation of Fitted Q-iteration

We now describe how the approximate value iteration algorithm can be easily implemented in the case when Q-functions are used (instead of value functions) and a generative model assumption is made.

We recall that a generative model is a simulator which receives as input a state and action (x, a) and it returns the corresponding reward $r(x, a)$ and a next state generated from $p(\cdot|x, a)$:



We recall that the optimal Q-function is defined by the Bellman equation

$$Q^*(x, a) = \sum_y p(y|x, a)\big[r(x, a, y) + \gamma V^*(y)\big].$$

and its the unique fixed point of the corresponding optimal Bellman operator $\mathcal{T}$ defined over $X \times A$ as:

$$\mathcal{T}Q(x, a) = \sum_y p(y|x, a)[r(x, a, y) + \gamma \max_b Q(y, b)].$$

As for the general approximate value iteration, the fitted Q-iteration algorithm can be represented as

$$Q_{k+1} = \mathcal{A}\mathcal{T}Q_k,$$

where $\mathcal{A}$ is an approximation operator over functions defined in $X \times A$. Unlike AVI, in this case, each iteration reduces exactly to the solution of a regression problem. We provide two examples, depending on the approximation scheme.

Linear approximation. Let $\mathcal{F}$ be a vector space over $X \times A$ defined by a set of d features $\phi_1, \dots, \phi_d : X \times A \to \mathbb{R}$, such that all the Q-functions in $\mathcal{F}$ can be represented as a linear combination of d features and weights α. In particular, we have

$$\mathcal{F} = \Big\{ Q_\alpha(x,a) = \sum_{j=1}^{d} \alpha_j \phi_j(x,a), \alpha \in \mathbb{R}^d \Big\}.$$

Let μ a distribution over X. We define the AVI scheme using $\mathcal{A}$ as the projection in $L_{2,\mu}$-norm onto the space $\mathcal{F}$. Thus, at each iteration we should solve the problem:

$$Q_{k+1} = \arg\min_{Q \in \mathcal{F}} \|Q - \mathcal{T} Q_k\|_\mu^2.$$

This introduces the first source of approximation due to the function space $\mathcal{F}$. Nonetheless, in practice we cannot compute exactly the operator $\mathcal{T}$ and we cannot take a minimization over the weighted norm in μ. Thus we need to introduce another source of approximation coming from sampling a finite number of points from μ and using the generative model to approximate $\mathcal{T}$. In particular, we build as sequence of functions $Q_k : X \times A \to \mathbb{R}$ such that at each iteration k

1. We first sample n state actions (X_i, A_i) where $X_i \sim \mu$ and A_i is chosen uniformly at random. Then we apply the generative model to the (X_i, A_i) pair to obtains (R_i, Y_i) as $Y_i \sim p(\cdot|X_i, A_i)$ and $R_i = r(X_i, A_i, Y_i)$,
2. We compute an estimation of $\mathcal{T} Q_k(X_i, A_i)$ as $Z_i = R_i + \gamma \max_{a \in A} Q_k(Y_i, a)$ (which is unbiased since $\mathbb{E}[Z_i | X_i, A_i] = \mathcal{T} Q_k(X_i, A_i)$),
3. We compute Q_{k+1} solving

$$Q_{k+1} = \arg\min_{Q_\alpha \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \big[Q_\alpha(X_i, A_i) - Z_i \big]^2. \tag{4}$$

   Since Q_α is a linear function of α, the problem is a simple quadratic minimization problem which can be solved in closed form by solving a linear system of equations of order d (number of features).

k-nearest neighbor. Before starting the actual algorithm we first

1. Sample n states $X_i \sim \mu$,
2. For each action a, we use the generative model to generate the next state $Y_{i,a}$ and the reward $R_{i,a}$ from (X_i, a)

If we compute the Q-function on the $n \times |A|$ sampled points, we can generalize it to any other state-action pair (x, a) using the value of the k closest points, that is

$$Q(x,a) = \frac{1}{k} \sum_{i=1}^{k} Q(X_{i(x)}, a), \tag{5}$$

where $i(x)$ is the index of the i-th closest state to x on the grid $\{X_i, 1 \le i \le n\}$.

Using this approximation scheme, we define the fitted Q-iteration procedure by first computing from Q_k the next iteration Q_{k+1} only on the points in the grid as

$$Q_{k+1}(X_i, a) = R_{i,a} + \gamma \max_{b \in A} Q_k(Y_{i,a}, b),$$

which then defines a function over $X \times A$ using the generalization from eq.(5).

Remark: the number of points k is critical since a very small k leads to **overfitting**, while a very big k leads to **bias**.

Remark: instead of using a fixed number of neighbours k, we can introduce a **kernel** $K(\cdot, \cdot)$ measuring the distance between states and generalizing the function as

$$Q(x, a) = \sum_{i=1}^{n} \frac{k(x, x_i)}{\sum_{j=1}^{n} k(x, x_j)} Q(X_i, a).$$

Other regression methods. Since fitted Q-iteration (unlike AVI) cast the iterative process as a sequence of regression problems, a wide range of options is available. The most popular regression methods for fitted Q-iteration are: regularized linear regression (with L_2 or L_1 regularization), non-linear regression with wavelets, neural networks, support vector machines, kernel methods (RKHS).

Problem: what are the theoretical guarantees for these algorithms when only a finite number of samples is available? *See next lecture...*

4.3 Example: the Optimal Replacement Problem

State: level of wear of an object (e.g., a car).

Action: {(R)eplace, (K)eep}.

Cost: $c(x, R) = C$, $c(x, K) = c(x)$ corresponding to the maintenance cost plus extra costs.

Dynamics: $p(\cdot|x, R) = \exp(\beta)$ with density $d(y) = \beta \exp^{-\beta y} \mathbb{I}\{y \geq 0\}$, $p(\cdot|x, K) = x + \exp(\beta)$ with density $d(y - x)$.

Problem: Minimize the discounted expected cost over an infinite horizon.

Recall that the optimal value function can be expressed using the optimal Bellman equation as

$$V^*(x) = \min \left\{ c(x) + \gamma \int_0^\infty d(y - x) V^*(y) dy,\ C + \gamma \int_0^\infty d(y) V^*(y) dy \right\}$$

where the optimal policy $\pi^*(x)$ is just the action which attains the minimum in the previous equation.

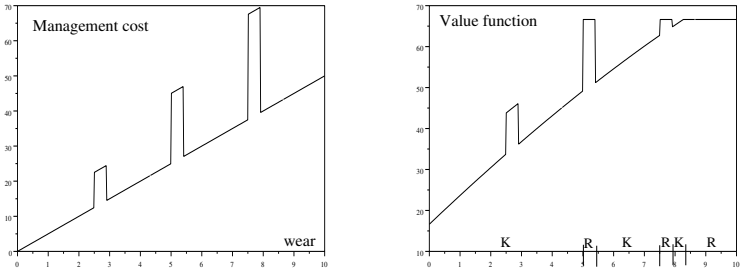


Figure 1: The management cost and the optimal value function.

We use the parameters $\gamma = 0.6$, $\beta = 0.6$ and $C = 50$, the cost function $c(x)$ and the optimal value function $V^*(x)$ are depicted in Figure 1.

Linear approximation. We define the approximation space as $\mathcal{F} := \left\{ V_n(x) = \sum_{k=1}^{20} \alpha_k \cos(k\pi \frac{x}{x_{\max}}) \right\}$. We collect the samples from a uniform grid of N points over the state space and we set the initial value function as $V_0 = 0$.

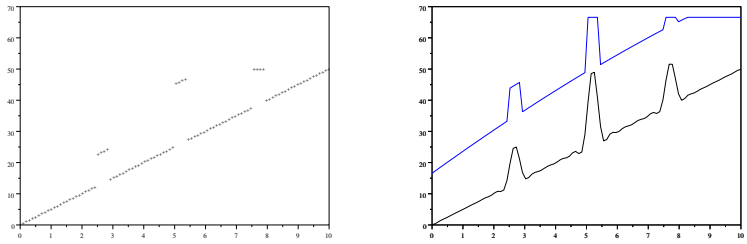


Figure 2: Left: the *target* values computed as $\{\mathcal{T}V_0(x_n)\}_{1 \le n \le N}$. Right: the approximation $V_1 \in \mathcal{F}$ of the target function $\mathcal{T}V_0$.

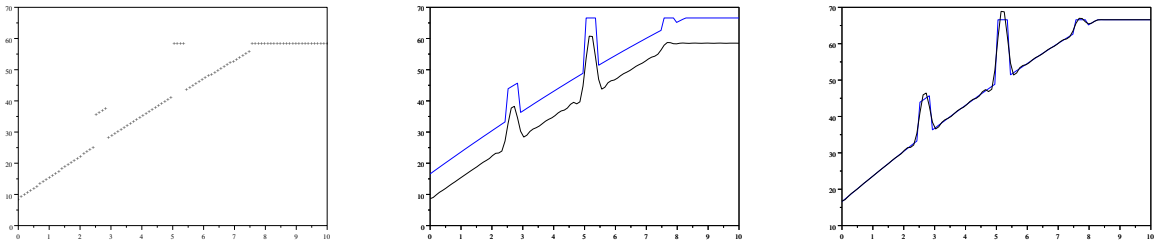


Figure 3: Left: the *target* values computed as $\{\mathcal{T}V_1(x_n)\}_{1 \le n \le N}$. Center: the approximation $V_2 \in \mathcal{F}$ of $\mathcal{T}V_1$. Right: the approximation $V_n \in \mathcal{F}$ after n iterations.

The approximate value iteration algorithm computes $\mathcal{T}V_0$ at the sampled points, it approximate it using functions in $\mathcal{F}$, and it reiterates the process. In Figure 4.3 we see the first iteration of this process and in Figure 4.3 the successive iterations.

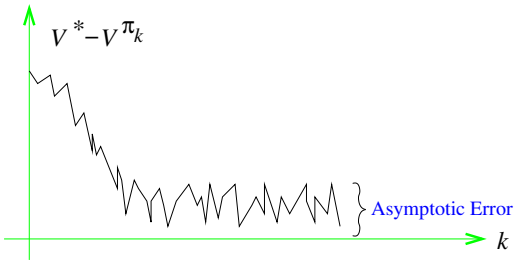


Figure 4: Illustration of the performance error over time for approximate policy iteration.

5 Approximate Policy Iteration

We recall that policy iteration is composed of two main steps. Let π_0 be an arbitrary policy, at each iteration k we have

- *Policy evaluation:* given the current policy π_k, compute V^{π_k}
- *Policy improvement:* given the value of the current policy, compute the greedy policy w.r.t. V^{π_k} as

$$\pi_{k+1}(x) \in \arg\max_{a \in A} \left[r(x,a) + \gamma \sum_{y \in X} p(y|x,a) V^{\pi_k}(y) \right].$$

We now focus on the case when this process cannot be done exactly but is only approximated. In particular, we focus on the case when the policy evaluation step suffers from an approximation error, and we obtain the **approximate policy iteration** algorithm.

Algorithm Definition 2. The **approximate policy iteration** (API) proceeds through iterations such that given an initial policy π_0, at each iteration k two steps are executed.

- *Policy evaluation:* given the current policy π_k, **approximate** its value V^{π_k} with a function V_k
- *Policy improvement:* given the **approximated** value of the current policy, compute the greedy policy w.r.t. V_k as

$$\pi_{k+1}(x) \in \arg\max_{a \in A} \left[r(x,a) + \gamma \sum_{y \in X} p(y|x,a) V_k(y) \right].$$

Problem: this algorithm is no longer guaranteed to converge, since we are not guaranteed that the policy improvement (run on the approximated V_k) does actually return a policy which is better than the previous one). Thus we study the asymptotic performance of the policies generated over iterations as in Figure 4

5.1 Approximation Error

Intuitively speaking, we expect that if the approximation error $\|V_k - V^{\pi_k}\|$ at each iteration is small, then the performance error should also be small, as proved in the next Proposition.

Proposition 5. The asymptotic performance of the policies π_k generated by the API algorithm is related to the approximation error as:

$$\limsup_{k\to\infty} \|V^* - V^{\pi_k}\|_\infty \leq \frac{2\gamma}{(1-\gamma)^2} \limsup_{k\to\infty} \|V_k - V^{\pi_k}\|_\infty$$

Proof. We introduce three different elements which are crucial to describe the performance of API:

- *Approximation error*: $e_k = V_k - V^{\pi_k}$,
- *Performance gain*: $g_k = V^{\pi_{k+1}} - V^{\pi_k}$,
- *Performance loss*: $l_k = V^* - V^{\pi_k}$.

Since π_{k+1} is greedy w.r.t. V_k we have that $T^{\pi_{k+1}} V_k \geq T^{\pi_k} V_k$. Thus we derive the following sequence of inequalities (component-wise)

$$\begin{aligned} g_k &= T^{\pi_{k+1}} V^{\pi_{k+1}} - T^{\pi_{k+1}} V^{\pi_k} + T^{\pi_{k+1}} V^{\pi_k} - T^{\pi_{k+1}} V_k + T^{\pi_{k+1}} V_k - T^{\pi_k} V_k + T^{\pi_k} V_k - T^{\pi_k} V^{\pi_k} \\ &\overset{(a)}{\geq} \gamma P^{\pi_{k+1}} g_k - \gamma (P^{\pi_{k+1}} - P^{\pi_k})\, e_k \\ &\overset{(b)}{\geq} -\gamma (I - \gamma P^{\pi_{k+1}})^{-1} (P^{\pi_{k+1}} - P^{\pi_k})\, e_k \end{aligned}$$

(a) Definition of e_k, g_k, and $\mathcal{T}^{\pi_k}$.

(b) Reordering.

This leads to the guarantee that

$$g_k \geq -\gamma (I - \gamma P^{\pi_{k+1}})^{-1} (P^{\pi_{k+1}} - P^{\pi_k})\, e_k, \tag{6}$$

which can be interpreted as the fact that the new policy cannot be much *worse* than the previous one. Although this does not correspond to the monotonic improvement we have in the exact policy iteration, it guarantees that the performance either improve or does not decrease much and that this is strictly related to the approximation error e_k.

Now we need to define a relationship between the performance at subsequent iterations. Since $T^{\pi^*} V_k \leq T^{\pi_{k+1}} V_k$ we have

$$\begin{aligned} l_{k+1} &= T^{\pi^*} V^* - T^{\pi^*} V^{\pi_k} + T^{\pi^*} V^{\pi_k} - T^{\pi^*} V_k \\ &\quad + T^{\pi^*} V_k - T^{\pi_{k+1}} V_k + T^{\pi_{k+1}} V_k - T^{\pi_{k+1}} V^{\pi_k} \\ &\quad + T^{\pi_{k+1}} V^{\pi_k} - T^{\pi_{k+1}} V^{\pi_{k+1}} \\ &\leq \gamma [P^{\pi^*} l_k - P^{\pi_{k+1}} g_k + (P^{\pi_{k+1}} - P^{\pi^*}) e_k]. \end{aligned}$$

If we now plug-in equation (6),

$$\begin{aligned} l_{k+1} &\leq \gamma P^{\pi^*} l_k + \gamma [P^{\pi_{k+1}} (I - \gamma P^{\pi_{k+1}})^{-1} (P^{\pi_{k+1}} - P^{\pi_k}) + P^{\pi_{k+1}} - P^{\pi^*}] e_k \\ &\leq \gamma P^{\pi^*} l_k + \gamma [P^{\pi_{k+1}} (I - \gamma P^{\pi_{k+1}})^{-1} (I - \gamma P^{\pi_k}) - P^{\pi^*}] e_k. \end{aligned}$$

Thus we obtain the fact that the performance loss changes through iterations as

$$l_{k+1} \quad \leq \quad \gamma P^{\pi^*} l_k + \gamma [P^{\pi_{k+1}} (I - \gamma P^{\pi_{k+1}})^{-1} (I - \gamma P^{\pi_k}) - P^{\pi^*}] e_k.$$

Now we need to study the asymptotic regime. Let $f_k = \gamma[P^{\pi_{k+1}}(I-\gamma P^{\pi_{k+1}})^{-1}(I-\gamma P^{\pi_k}) - P^{\pi^*}]e_k$, we have

$$l_{k+1} \leq \gamma P^{\pi^*} l_k + f_k,$$

thus if we move to the lim sup we obtain,

$$\begin{aligned}(I-\gamma P^{\pi^*}) \limsup_{k\to\infty} l_k &\leq \limsup_{k\to\infty} f_k \\ \limsup_{k\to\infty} l_k &\leq (I-\gamma P^{\pi^*})^{-1} \limsup_{k\to\infty} f_k,\end{aligned}$$

since $I-\gamma P^{\pi^*}$ is invertible. Finally, we only need to take the L_∞-norm both sides and obtain,

$$\begin{aligned}\limsup_{k\to\infty} \|l_k\| &\leq \frac{\gamma}{1-\gamma} \limsup_{k\to\infty} \|P^{\pi_{k+1}}(I-\gamma P^{\pi_{k+1}})^{-1}(I+\gamma P^{\pi_k}) + P^{\pi^*}\| \, \|e_k\| \\ &\leq \frac{\gamma}{1-\gamma}\left(\frac{1+\gamma}{1-\gamma}+1\right) \limsup_{k\to\infty} \|e_k\| = \frac{2\gamma}{(1-\gamma)^2} \limsup_{k\to\infty} \|e_k\|.\end{aligned}$$

□

5.2 Policy Evaluation with Linear Approximation

We now consider a specific approximation scheme for the policy evaluation step using a linear approximation. In particular, we approximate the value function V^π using a vector space $\mathcal{F}$ defined by the features $\phi_1, \ldots, \phi_d : X \to \mathbb{R}$:

$$\mathcal{F} = \{V_\alpha(x) = \sum_{i=1}^{d} \alpha_i \phi_i(x), \alpha \in \mathbb{R}^d\}.$$

The objective is to obtain the parameter $\alpha \in \mathbb{R}^d$ such that V_α is a good approximation of V^π.

5.2.1 Extension of TD(λ) to Linear Approximation

The algorithm. The TD(λ) algorithm is the same that we defined in Lecture 3, adjusted for a linear approximation, so that the parameter vector α is updated according to the (approximated) temporal difference.

Algorithm Definition 3. We define a trace vector $z \in \mathbb{R}^d$ (same size as α) initialized to zero. Starting from an initial state x_0 we generate a sequence of states $(x_0, x_1, x_2, \ldots)$ choosing actions according to the policy π under evaluation. At each step t, we compute the temporal difference according to the current approximation V_α, that is:

$$d_t = r(x_t, \pi(x_t)) + \gamma V_\alpha(x_{t+1}) - V_\alpha(x_t)$$

and we use it to update both the parameter vector and the trace vector as:

$$\begin{aligned}\alpha_{t+1} &= \alpha_t + \eta_t d_t z_t, \\ z_{t+1} &= \lambda\gamma z_t + \phi(x_{t+1}),\end{aligned}$$

where η_t is learning step and $\phi : X \to \mathbb{R}^d$ is a vector function with elements ϕ_i.

Building on the results from stochastic approximation, we have that the sequence α_{t+1} actually converges and a performance guarantee can be derived for the value function returned at convergence.

Approximation error.

Proposition 6 (Tsitsiklis et Van Roy, 1996). Let the learning rate η_t satisfy

$$\sum_{t\geq 0} \eta_t = \infty, \text{ and } \sum_{t\geq 0} \eta_t^2 < \infty.$$

We assume that there exists a distribution μ over X such that $\forall x, x' \in X$, $\lim_{t\to\infty} P(x_t = x'|x_0 = x) = \mu(x')$ and that the features $(\phi_i)_{1\leq k\leq K}$ are linearly independent. Then there exists a fixed α^* such that

$$\lim_{t\to\infty} \alpha_t = \alpha^*.$$

Furthermore we obtain

$$\|V_{\alpha^*} - V^\pi\|_{2,\mu} \leq \frac{1-\lambda\gamma}{1-\gamma} \inf_\alpha \|V_\alpha - V^\pi\|_{2,\mu}. \tag{7}$$

Remark: for $\lambda = 1$, we recover the Monte-Carlo (or TD(1)) algorithm and the approximation coincides with the best possible approximation of V^π in the space $\mathcal{F}$ (i.e., the project of V^π onto $\mathcal{F}$). As λ decreases, we obtain worse and worse approximation (due to the bias of the approximation) but we have an estimator with much smaller variance, which allows α_t to converge to α^* much faster.

Implementation. The implementation of TD(λ) is fully online, so that at each step t, the vector α is updated using the temporal difference d_t, but then this sample is forgotten and never used again. This makes TD(λ) a not very sample efficient algorithm, since it requires a large number of steps before converging to α^*. In particular, it requires the same state x to be visited many times before having a stable value of $V_\alpha(x)$. This leads to the definition of the more data-efficient version of TD introduced in the next section.

5.2.2 Least Squares Temporal Difference

The algorithm. In the definition of the least squares temporal difference (LSTD) algorithm, we focus on the fact that V^π is the fixed point of the Bellman operator T^π. The objective is to learns the function in $\mathcal{F}$ which better approximate V^π w.r.t. a given norm $\|\cdot\|$.

As commented before, we could try to use a projection Π_∞ in norm L_∞ and exploit the fact that the operator $\Pi_\infty \mathcal{T}^\pi$ is a contraction in the L_∞-norm. Nonetheless, the Π_∞ is not numerically feasible when the number of states N increases. Thus we rely on different norms such as L_2 (e.g., linear regression, neural networks), or L_1 (e.g., SVM, Lasso).

In the following we focus on the $L_{2,\mu}$ weighted norm with μ a distribution over X and the corresponding projection Π_μ defined as:

$$\Pi_\mu g = \arg\min_{f\in\mathcal{F}} \|f - g\|_\mu.$$

Thus the idea is to compute the fixed point of the joint operator $\Pi_\mu T^\pi$. If this operator admits a fixed point, then we call it the LSTD solution and we denote it by V_{TD}. Using the illustration in Figure 5 it is clear that the Bellman residual corresponding to V_{TD} must be orthogonal to the approximation space $\mathcal{F}$. In particular, we have that $\mathcal{T}^\pi V_{TD} - V_{TD} \perp \mathcal{F}$. This implies that for any $1 \leq i \leq d$

$$\langle \mathcal{T}^\pi V_{TD} - V_{TD}, \phi_i \rangle_\mu = 0,$$

where the scalar product is defined as $\langle f, g\rangle_\mu = \sum_{x\in X} f(x)g(x)\mu(x)$. Further elaborating on the previous

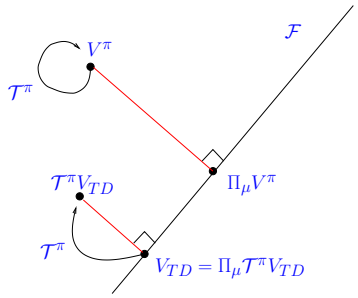


Figure 5: Illustration of the LSTD solutions and the projection of V^π onto $\mathcal{F}$.

condition, we have that for any $1 \leq i \leq d$:

$$\begin{aligned} \langle r^\pi + \gamma P^\pi V_{TD} - V_{TD}, \phi_i \rangle_\mu &= 0 \\ \langle r^\pi, \phi_i \rangle_\mu + \sum_{j=1}^{d} \langle \gamma P^\pi \phi_j - \phi_j, \phi_i \rangle_\mu \alpha_{TD,j} &= 0, \end{aligned}$$

which implies that α_{TD} can be computed as the solution of a linear system of order d.

Algorithm Definition 4. The LSTD solution α_{TD} can be computed by computing the matrix A and vector b defined as

$$\begin{aligned} A_{i,j} &= \langle \phi_i, \phi_j - \gamma P^\pi \phi_j \rangle_\mu \\ b_i &= \langle \phi_i, r^\pi \rangle_\mu \end{aligned}, \tag{8}$$

and then solving the system $A\alpha = b$.

Approximation error. The main problem with LSTD is that in general $\Pi_\mu \mathcal{T}^\pi$ may not admit a fixed point and even when it does, it is not trivial to provide a bound on its approximation error. In fact, depending on the choice of μ, we might have that the matrix A is not invertible.

Thus, we only focus on the case (similar to TD) when the distribution μ coincides with the stationary distribution μ_π defined by the current policy π, that is

$$\mu_\pi P^\pi = \mu_\pi, \text{ and } \mu_\pi(y) = \sum_x p(y|x, \pi(x)) \mu_\pi(x)$$

for any $y \in X$. Notice that fact that μ_π exists is itself an assumption on the Markov chain generated by following π in the MDP at hand. Then we can indeed guarantee that the joint operator $\Pi_{\mu_\pi} \mathcal{T}^\pi$ has a unique fixed point and derive the following guarantee.

Proposition 7. Let π admit a stationary distribution μ_π. Then the Bellman operator $\mathcal{T}^\pi$ is a contraction in the weighted L_{2,μ_π}-norm. Thus the joint operator $\Pi_{\mu_\pi} \mathcal{T}^\pi$ is a contraction and it admits a unique fixed point V_{TD}. Then the following approximation error guarantee holds:

$$\|V^\pi - V_{TD}\|_{\mu_\pi} \leq \frac{1}{\sqrt{1-\gamma^2}} \inf_{V \in \mathcal{F}} \|V^\pi - V\|_{\mu_\pi}. \tag{9}$$

Proof. We first show that the transition matrix of the Markov chain induced by π is such that $\|P_\pi\|_{\mu_\pi} = 1$.

In fact, we have that

$$\begin{aligned}\|P^\pi V\|^2_{\mu_\pi} &= \sum_x \mu_\pi(x)\big(\sum_y p(y|x,\pi(x))V(y)\big)^2 \\ &\leq \sum_x \sum_y \mu_\pi(x)p(y|x,\pi(x))V(y)^2 \\ &= \sum_y \mu_\pi(y)V(y)^2 = \|V\|^2_{\mu_\pi}.\end{aligned}$$

Then it immediately follows that $\mathcal{T}^\pi$ is a contraction in L_{2,μ_π}, i.e.,

$$\|\mathcal{T}^\pi V_1 - \mathcal{T}^\pi V_2\|_{\mu_\pi} = \gamma\|P^\pi(V_1 - V_2)\|_{\mu_\pi} \leq \gamma\|V_1 - V_2\|_{\mu_\pi}.$$

Furthermore, we can show that Π_{μ_π} is a non-expansion, thus the composition of a non-expansion and a contraction leads to the joint operator $\Pi_{\mu_\pi}\mathcal{T}^\pi$ which is still a γ contraction in L_{2,μ_π} with unique fixed point $V_{TD} = \Pi_{\mu_\pi}\mathcal{T}^\pi V_{TD}$. By Pythagorean theorem we have

$$\|V^\pi - V_{TD}\|^2_{\mu_\pi} = \|V^\pi - \Pi_{\mu_\pi}V^\pi\|^2_{\mu_\pi} + \|\Pi_{\mu_\pi}V^\pi - V_{TD}\|^2_{\mu_\pi},$$

but

$$\|\Pi_{\mu_\pi}V^\pi - V_{TD}\|^2_{\mu_\pi} = \|\Pi_{\mu_\pi}V^\pi - \Pi_{\mu_\pi}\mathcal{T}^\pi V_{TD}\|^2_{\mu_\pi} \leq \|\mathcal{T}^\pi V^\pi - \mathcal{T}V_{TD}\|^2_{\mu_\pi} \leq \gamma^2\|V^\pi - V_{TD}\|^2_{\mu_\pi}.$$

Thus

$$\|V^\pi - V_{TD}\|^2_{\mu_\pi} \leq \|V^\pi - \Pi_{\mu_\pi}V^\pi\|^2_{\mu_\pi} + \gamma^2\|V^\pi - V_{TD}\|^2_{\mu_\pi},$$

which corresponds to eq.(9) after reordering. □

Implementation. Similar to other algorithms, the previous analysis is done only on the approximation error coming from using a restricted linear space $\mathcal{F}$. Nonetheless, the computation of matrix A still requires the computation of the application of the transition P to the features and vector b the full knowledge of the reward function. In general, this model is not available, so we need to rely on samples directly generated from the environment. Unlike fitted Q-iteration, here we do not need a generative model but we only need a single trajectory $(X_0, X_1, \dots)$ obtained by executing the policy π in the environment, so that $X_{t+1} \sim p(\cdot|X_t, \pi(X_t)))$. Let $R_t = r(X_t, \pi(X_t))$ be the reward observed at each time step t, we can construct estimates of the matrix A and vector b (see eq.(8)) as

$$\begin{aligned}\hat{A}_{ij} &= \frac{1}{n}\sum_{t=1}^n \phi_i(X_t)[\phi_j(X_t) - \gamma\phi_j(X_{t+1})], \\ \hat{b}_i &= \frac{1}{n}\sum_{t=1}^n \phi_i(X_t)R_t.\end{aligned}$$

and we compute the (approximate) LSTD solution by solving $\hat{A}\alpha = \hat{b}$. If the Markov chain is ergodic then the empirical distribution of the states in the trajectory (X_t) tends towards the stationary distribution. Thus we have the guarantee that when the length of the trajectory tends to infinity ($n \to \infty$) we have that $\hat{A} \to A$ et $\hat{b} \to b$ when $n \to \infty$.

Problem: what happens when the number of samples in indeed finite? See Lecture 6.

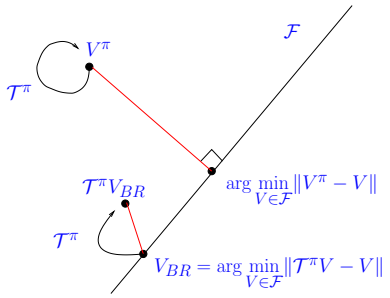


Figure 6: Illustration of the Bellman residual minimization algorithm.

5.2.3 Bellman Residual Minimization (BRM)

The Algorithm. Similar to Section 3, we use the notion of Bellman residual to choose the best approximation of V^π in $\mathcal{F}$. In particular, we study the Bellman residual no longer w.r.t. the optimal Bellman operator but w.r.t. the Bellman operator defined by the current policy $\mathcal{T}^\pi$. The objective is to compute

$$V_{BR} = \arg\min_{V\in\mathcal{F}} \|T^\pi V - V\|, \tag{10}$$

for a given norm $\|\cdot\|$, as illustrated in Figure 6.

Let μ be an arbitrary distribution over X, we denote by V_{BR} the minimum of the Bellman residual in eq.(10) with norm $L_{2,\mu}$. Unlike in the case of BRM for the optimal value function, we notice that the mapping $\alpha \to \mathcal{T}^\pi V_\alpha - V_\alpha$ is affine and thus the function $\alpha \to \|\mathcal{T}^\pi V_\alpha - V_\alpha\|_\mu^2$ is quadratic. The minimum of this function can be achieved by computing the gradient and setting it to zero, thus obtaining the linear system

$$\langle r^\pi + (\gamma P^\pi - I)\sum_{j=1}^{d} \phi_j \alpha_j, (\gamma P^\pi - I)\phi_i\rangle_\mu = 0, \quad \text{for any } 1 \le i \le d,$$

which can be rewritten as

$$A\alpha = b,$$

with matrix A and vector b defined as

$$\begin{cases} A_{i,j} &= \langle \phi_i - \gamma P^\pi \phi_i, \phi_j - \gamma P^\pi \phi_j\rangle_\mu, & \text{for } 1 \le i,j \le d \\ b_i &= \langle \phi_i - \gamma P^\pi \phi_i, r^\pi\rangle_\mu, & \text{for } 1 \le i \le d \end{cases} \tag{11}$$

This system always admits a solution when the features ϕ_i are linearly independent w.r.t. the chosen distribution μ. Unlike the LSTD linear system, this does not immediately coincide with the solution of linear regression, but we can notice that if we introduce a new basis $\{\psi_i = \phi_i - \gamma P^\pi \phi_i\}_{i=1\dots d}$, then the previous system can be interpreted as the minimization of the linear regression problem $\|\alpha \cdot \psi - r^\pi\|_\mu$, which suggests that standard supervised learning techniques could be employed.

Approximation error. We can derive an approximation error which relates the quality of V_{BR} to the best approximation of V^π in $\mathcal{F}$.

Proposition 8. we have that

$$\|V^\pi - V_{BR}\| \le \|(I - \gamma P^\pi)^{-1}\|(1 + \gamma\|P^\pi\|) \inf_{V\in\mathcal{F}} \|V^\pi - V\|. \tag{12}$$

Furthermore if μ_π is the stationary policy of π, then $\|P^\pi\|_{\mu_\pi} = 1$ and $\|(I-\gamma P^\pi)^{-1}\|_{\mu_\pi} = \frac{1}{1-\gamma}$, thus

$$\|V^\pi - V_{BR}\|_{\mu_\pi} \leq \frac{1+\gamma}{1-\gamma} \inf_{V\in\mathcal{F}} \|V^\pi - V\|_{\mu_\pi}.$$

Proof. For any function V we can relate the Bellman residual to the approximation error as

$$\begin{aligned} V^\pi - V &= V^\pi - T^\pi V + T^\pi V - V = \gamma P^\pi (V^\pi - V) + T^\pi V - V \\ (I-\gamma P^\pi)(V^\pi - V) &= T^\pi V - V, \end{aligned}$$

taking the norm both sides we obtain

$$\|V^\pi - V_{BR}\| \leq \|(I-\gamma P^\pi)^{-1}\| \|\mathcal{T}^\pi V_{BR} - V_{BR}\|$$

and

$$\|\mathcal{T}^\pi V_{BR} - V_{BR}\| = \inf_{V\in\mathcal{F}} \|\mathcal{T}^\pi V - V\| \leq (1+\gamma\|P^\pi\|) \inf_{V\in\mathcal{F}} \|V^\pi - V\|,$$

from which we deduce eq.(12).

If we consider the stationary distribution μ_π, we have that $\|P^\pi\|_{\mu_\pi} = 1$ and we can use the fact that P^π is a stochastic matrix and that the inverse of $(I-\gamma P^\pi)$ can be written as the power series $\sum_t \gamma (P^\pi)^t$. Applying the norm we obtain that $\|(I-\gamma P^\pi)^{-1}\|_{\mu_\pi} \leq \sum_{t\geq 0} \gamma^t \|P^\pi\|^t_{\mu_\pi} \leq \frac{1}{1-\gamma}$.

□

Implementation. We first study the implementation of the Bellman residual minimization in the general case of an arbitrary distribution μ. We assume access to a generative model is available. We need to compute an estimator of the Bellman residual

$$\mathcal{B}(V) = \|\mathcal{T}^\pi V - V\|^2_\mu.$$

We first drawn n states from the distribution μ $X_t \sim \mu$ and we call the generative model with input (X_t, A_t) (with $A_t = \pi(X_t)$) and obtain the reward $R_t = r(X_t, A_t)$ and the next state $Y_t \sim p(\cdot|X_t, A_t)$. Thus we define the empirical estimator

$$\hat{\mathcal{B}}(V) = \frac{1}{n}\sum_{t=1}^n \big[V(X_t) - \underbrace{\big(R_t + \gamma V(Y_t)\big)}_{\hat{\mathcal{T}}V(X_t)}\big]^2.$$

Problem: this estimator is biased (and not consistent)! In fact,

$$\begin{aligned} \mathbb{E}[\hat{\mathcal{B}}(V)] &= \mathbb{E}\Big[\big[V(X_t) - \mathcal{T}^\pi V(X_t) + \mathcal{T}^\pi V(X_t) - \hat{\mathcal{T}}V(X_t)\big]^2\Big] \\ &= \|\mathcal{T}^\pi V - V\|^2_\mu + \mathbb{E}\Big[\big[\mathcal{T}^\pi V(X_t) - \hat{\mathcal{T}}V(X_t)\big]^2\Big] \end{aligned}$$

As a result, minimizing $\hat{\mathcal{B}}(V)$ does not correspond to minimizing $\mathcal{B}(V)$, even when $n \to \infty$.

We can solve this problem by generating multiple samples in each state X_t. In particular, in each state X_t, we generate two independent samples Y_t et $Y'_t \sim p(\cdot|X_t, A_t)$ and define the estimator

$$\hat{\mathcal{B}}(V) = \frac{1}{n}\sum_{t=1}^n \big[V(X_t) - \big(R_t + \gamma V(Y_t)\big)\big]\big[V(X_t) - \big(R_t + \gamma V(Y'_t)\big)\big].$$

Although this estimator now requires $2n$ calls to the generative mode, it returns an unbiased estimator of the Bellman residual, since $\mathbb{E}\hat{\mathcal{B}}(V) = \mathcal{B}(V)$. As a result, when $n \to \infty$, we have the guarantee that the minimum of $\hat{\mathcal{B}}$ will coincide with the minimum of $\mathcal{B}$.

Since the function $\alpha \to \hat{\mathcal{B}}(V_\alpha)$ is still quadratic, we can still reformulate the previous problem as the solution of a linear system with

$$\begin{aligned}\widehat{A}_{i,j} &= \frac{1}{n}\sum_{t=1}^{n}\big[\phi_i(X_t) - \gamma\phi_i(Y_t)\big]\big[\phi_j(X_t) - \gamma\phi_j(Y_t')\big],\\ \widehat{b}_i &= \frac{1}{n}\sum_{t=1}^{n}\big[\phi_i(X_t) - \gamma\frac{\phi_i(Y_t)+\phi_i(Y_t')}{2}\big]\,R_t.\end{aligned}$$

5.2.4 Pros and cons of LSTD and BRM

- **Different assumptions:** BRM requires a generative model, while LSTD only requires a single trajectory.
- **The performance is evaluated differently:** The approximation error $\|V^\pi - \hat{V}\|_\mu$ is measure w.r.t. the sampling distribution μ used to collect the samples. While BRM allows to use any possible sampling distribution, LSTD is strictly bound to use the stationary distribution μ_π. As a result, V^π might be very poorly approximated by V_{TD} in the regions of the states space which are not covered by the policy and this might result in a very poor performance once moving to the policy improvement step.

5.3 Policy Improvement

If working with value functions, the policy improvement step, after computing the approximation V_k requires to compute a policy π_{k+1} defined as

$$\pi_{k+1}(x) \in \arg\max_{a\in A}[r(x,a) + \gamma\sum_y p(y|x,a)V_k(y)].$$

This is often impossible since the reward r and the dynamics p are not known. As a result, here an additional approximation might be needed. Fortunately, we can find a workaround for this problem by using Q-functions instead of value functions. In fact, if an approximation Q_k is provided, then the policy improvement is simply

$$\pi_{k+1}(x) \in \arg\max_{a\in A} Q_k(x,a).$$

Let first define a vector space $\mathcal{F}$ defined over $X \times A$ using features $\phi_1, \dots, \phi_d : X \times A \to \mathbb{R}$:

$$\mathcal{F} = \{Q_\alpha(x,a) = \sum_{j=1}^{d}\alpha_j\phi_j(x,a), \alpha \in \mathbb{R}^d\}.$$

LSTD Algorithm: We generate a trajectory $(X_0, X_1, \dots)$ following the policy π_k (i.e., $X_{t+1} \sim p(\cdot|X_t, \pi_k(X_t))$). Let $R_t = r(X_t, \pi_k(X_t))$, then the matrix $\hat{A}$ and the vector $\hat{b}$ can be computed as

$$\begin{aligned}\hat{A}_{ij} &= \frac{1}{n}\sum_{t=1}^{n}\phi_i(X_t,A_t)[\phi_j(X_t,A_t) - \gamma\phi_j(X_{t+1},A_{t+1})],\\ \hat{b}_i &= \frac{1}{n}\sum_{t=1}^{n}\phi_i(X_t,A_t)R_t.\end{aligned}$$

Then we solve the system $\hat{A}\alpha = \hat{b}$ and we obtain $\hat{\alpha}_{TD}$.

BRM Algorithm: We generate $n$states $X_t \sim \mu$, $A_t = \pi_k(X_t)$ and we simulate using a generative model the rewards $R_t = r(X_t, A_t)$ and a pair of next states Y_t from $Y_t' \sim p(\cdot|X_t, A_t)$. Let $B_t = \pi_k(Y_t)$ and $B_t' = \pi_k(Y_t')$, then the matrix $\hat{A}$ and the vector $\hat{b}$ can be computed as

$$\begin{aligned}
\widehat{A}_{i,j} &= \frac{1}{n}\sum_{t=1}^{n}\big[\phi_i(X_t, A_t) - \gamma\phi_i(Y_t, B_t)\big]\big[\phi_j(X_t, A_t) - \gamma\phi_j(Y_t', B_t')\big], \\
\widehat{b}_i &= \frac{1}{n}\sum_{t=1}^{n}\big[\phi_i(X_t, A_t) - \gamma\frac{\phi_i(Y_t, B_t) + \phi_i(Y_t', B_t')}{2}\big]\, R_t.
\end{aligned}$$

Then we solve the system $\hat{A}\alpha = \hat{b}$ and we obtain $\hat{\alpha}_{BR}$.