



Approximate Dynamic Programming

A. LAZARIC (*SequeL Team @INRIA-Lille*)
ENS Cachan - Master 2 MVA

SequeL – INRIA Lille

Approximate Dynamic Programming

(a.k.a. Batch Reinforcement Learning)

Approximate Dynamic Programming

(a.k.a. Batch Reinforcement Learning)

Approximate Value Iteration

Approximate Policy Iteration

From DP to ADP

- ▶ Dynamic programming algorithms require an *explicit* definition of
 - ▶ transition probabilities $p(\cdot|x, a)$
 - ▶ reward function $r(x, a)$

From DP to ADP

- ▶ Dynamic programming algorithms require an *explicit* definition of
 - ▶ transition probabilities $p(\cdot|x, a)$
 - ▶ reward function $r(x, a)$
- ▶ This knowledge is often *unavailable* (i.e., wind intensity, human-computer-interaction).

From DP to ADP

- ▶ Dynamic programming algorithms require an *explicit* definition of
 - ▶ transition probabilities $p(\cdot|x, a)$
 - ▶ reward function $r(x, a)$
- ▶ This knowledge is often *unavailable* (i.e., wind intensity, human-computer-interaction).
- ▶ *Can we rely on samples?*

From DP to ADP

- ▶ Dynamic programming algorithms require an *exact* representation of value functions and policies

From DP to ADP

- ▶ Dynamic programming algorithms require an *exact* representation of value functions and policies
- ▶ This is often *impossible* since their shape is too “complicated” (e.g., large or continuous state space).

From DP to ADP

- ▶ Dynamic programming algorithms require an *exact* representation of value functions and policies
- ▶ This is often *impossible* since their shape is too “complicated” (e.g., large or continuous state space).
- ▶ *Can we use approximations?*

The Objective

Find a policy π such that
the *performance loss* $\|V^* - V^\pi\|$ is as small as possible

From Approximation Error to Performance Loss

Question: if V is an approximation of the optimal value function V^* with an error

$$\text{error} = \|V - V^*\|$$

From Approximation Error to Performance Loss

Question: if V is an approximation of the optimal value function V^* with an error

$$\text{error} = \|V - V^*\|$$

how does it translate to the (loss of) performance of the *greedy policy*

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

From Approximation Error to Performance Loss

Question: if V is an approximation of the optimal value function V^* with an error

$$\text{error} = \|V - V^*\|$$

how does it translate to the (loss of) performance of the *greedy policy*

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

i.e.

$$\text{performance loss} = \|V^* - V^\pi\|$$

From Approximation Error to Performance Loss

Proposition

Let $V \in \mathbb{R}^N$ be an approximation of V^* and π its corresponding greedy policy, then

$$\underbrace{\|V^* - V^\pi\|_\infty}_{\text{performance loss}} \leq \frac{2\gamma}{1-\gamma} \underbrace{\|V^* - V\|_\infty}_{\text{approx. error}}.$$

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

From Approximation Error to Performance Loss

Proof.

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



Approximate Dynamic Programming

(a.k.a. Batch Reinforcement Learning)

Approximate Value Iteration

Approximate Policy Iteration

From Approximation Error to Performance Loss

Question: how do we compute a *good* V ?

From Approximation Error to Performance Loss

Question: how do we compute a *good* V ?

Problem: unlike in standard approximation scenarios (see supervised learning), we have a *limited access* to the target function, i.e. V^* .

From Approximation Error to Performance Loss

Question: how do we compute a *good* V ?

Problem: unlike in standard approximation scenarios (see supervised learning), we have a *limited access* to the target function, i.e. V^* .

Solution: value iteration tends to learn functions which are *close to the optimal value function* V^* .

Value Iteration: the Idea

1. Let Q_0 be *any* action-value function
2. At each iteration $k = 1, 2, \dots, K$

- ▶ Compute

$$Q_{k+1}(x, a) = \mathcal{T}Q_k(x, a) = r(x, a) + \sum_y p(y|x, a) \gamma \max_b Q_k(y, b)$$

3. Return the *greedy* policy

$$\pi_K(x) \in \arg \max_{a \in A} Q_K(x, a).$$

Value Iteration: the Idea

1. Let Q_0 be *any* action-value function
2. At each iteration $k = 1, 2, \dots, K$

- ▶ Compute

$$Q_{k+1}(x, a) = \mathcal{T}Q_k(x, a) = r(x, a) + \sum_y p(y|x, a) \gamma \max_b Q_k(y, b)$$

3. Return the *greedy* policy

$$\pi_K(x) \in \arg \max_{a \in A} Q_K(x, a).$$

- ▶ **Problem:** how can we approximate $\mathcal{T}Q_k$?
- ▶ **Problem:** if $Q_{k+1} \neq \mathcal{T}Q_k$, does (approx.) value iteration still work?

Linear Fitted Q-iteration: the Approximation Space

Linear space (used to approximate action–value functions)

$$\mathcal{F} = \left\{ f(x, a) = \sum_{j=1}^d \alpha_j \varphi_j(x, a), \quad \alpha \in \mathbb{R}^d \right\}$$

Linear Fitted Q-iteration: the Approximation Space

Linear space (used to approximate action–value functions)

$$\mathcal{F} = \left\{ f(x, a) = \sum_{j=1}^d \alpha_j \varphi_j(x, a), \quad \alpha \in \mathbb{R}^d \right\}$$

with features

$$\varphi_j : X \times A \rightarrow [0, L] \quad \phi(x, a) = [\varphi_1(x, a) \dots \varphi_d(x, a)]^\top$$

Linear Fitted Q-iteration: the Samples

Assumption: access to a **generative model**, that is a black-box simulator $\text{sim}()$ of the environment is available.

Given (x, a) ,

$$\text{sim}(x, a) = \{y, r\}, \quad \text{with } y \sim p(\cdot | x, a), r = r(x, a)$$

Linear Fitted Q-iteration

Input: space \mathcal{F} , iterations K , sampling distribution ρ , num of samples n

Linear Fitted Q-iteration

Input: space \mathcal{F} , iterations K , sampling distribution ρ , num of samples n

Initial function $\hat{Q}_0 \in \mathcal{F}$

Linear Fitted Q-iteration

Input: space \mathcal{F} , iterations K , sampling distribution ρ , num of samples n

Initial function $\hat{Q}_0 \in \mathcal{F}$

For $k = 1, \dots, K$

Linear Fitted Q-iteration

Input: space \mathcal{F} , iterations K , sampling distribution ρ , num of samples n

Initial function $\hat{Q}_0 \in \mathcal{F}$

For $k = 1, \dots, K$

1. Draw n samples $(x_i, a_i) \stackrel{\text{i.i.d}}{\sim} \rho$

Linear Fitted Q-iteration

Input: space \mathcal{F} , iterations K , sampling distribution ρ , num of samples n

Initial function $\hat{Q}_0 \in \mathcal{F}$

For $k = 1, \dots, K$

1. Draw n samples $(x_i, a_i) \stackrel{\text{i.i.d}}{\sim} \rho$
2. Sample $x'_i \sim p(\cdot | x_i, a_i)$ and $r_i = r(x_i, a_i)$

Linear Fitted Q-iteration

Input: space \mathcal{F} , iterations K , sampling distribution ρ , num of samples n

Initial function $\hat{Q}_0 \in \mathcal{F}$

For $k = 1, \dots, K$

1. Draw n samples $(x_i, a_i) \stackrel{\text{i.i.d}}{\sim} \rho$
2. Sample $x'_i \sim p(\cdot | x_i, a_i)$ and $r_i = r(x_i, a_i)$
3. Compute $y_i = r_i + \gamma \max_a \hat{Q}_{k-1}(x'_i, a)$

Linear Fitted Q-iteration

Input: space \mathcal{F} , iterations K , sampling distribution ρ , num of samples n

Initial function $\hat{Q}_0 \in \mathcal{F}$

For $k = 1, \dots, K$

1. Draw n samples $(x_i, a_i) \stackrel{\text{i.i.d}}{\sim} \rho$
2. Sample $x'_i \sim p(\cdot | x_i, a_i)$ and $r_i = r(x_i, a_i)$
3. Compute $y_i = r_i + \gamma \max_a \hat{Q}_{k-1}(x'_i, a)$
4. Build training set $\{(x_i, a_i), y_i\}_{i=1}^n$

Linear Fitted Q-iteration

Input: space \mathcal{F} , iterations K , sampling distribution ρ , num of samples n

Initial function $\hat{Q}_0 \in \mathcal{F}$

For $k = 1, \dots, K$

1. Draw n samples $(x_i, a_i) \stackrel{\text{i.i.d}}{\sim} \rho$
2. Sample $x'_i \sim p(\cdot | x_i, a_i)$ and $r_i = r(x_i, a_i)$
3. Compute $y_i = r_i + \gamma \max_a \hat{Q}_{k-1}(x'_i, a)$
4. Build training set $\{(x_i, a_i), y_i\}_{i=1}^n$
5. Solve the *least squares problem*

$$f_{\hat{\alpha}_k} = \arg \min_{f_{\alpha} \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n (f_{\alpha}(x_i, a_i) - y_i)^2$$

Linear Fitted Q-iteration

Input: space \mathcal{F} , iterations K , sampling distribution ρ , num of samples n

Initial function $\hat{Q}_0 \in \mathcal{F}$

For $k = 1, \dots, K$

1. Draw n samples $(x_i, a_i) \stackrel{\text{i.i.d}}{\sim} \rho$
2. Sample $x'_i \sim p(\cdot | x_i, a_i)$ and $r_i = r(x_i, a_i)$
3. Compute $y_i = r_i + \gamma \max_a \hat{Q}_{k-1}(x'_i, a)$
4. Build training set $\{(x_i, a_i), y_i\}_{i=1}^n$
5. Solve the *least squares problem*

$$f_{\hat{\alpha}_k} = \arg \min_{f_{\alpha} \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n (f_{\alpha}(x_i, a_i) - y_i)^2$$

6. Return $\hat{Q}_k = f_{\hat{\alpha}_k}$ (*truncation may be needed*)

Linear Fitted Q-iteration

Input: space \mathcal{F} , iterations K , sampling distribution ρ , num of samples n

Initial function $\hat{Q}_0 \in \mathcal{F}$

For $k = 1, \dots, K$

1. Draw n samples $(x_i, a_i) \stackrel{\text{i.i.d}}{\sim} \rho$
2. Sample $x'_i \sim p(\cdot | x_i, a_i)$ and $r_i = r(x_i, a_i)$
3. Compute $y_i = r_i + \gamma \max_a \hat{Q}_{k-1}(x'_i, a)$
4. Build training set $\{((x_i, a_i), y_i)\}_{i=1}^n$
5. Solve the *least squares problem*

$$f_{\hat{\alpha}_k} = \arg \min_{f_{\alpha} \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n (f_{\alpha}(x_i, a_i) - y_i)^2$$

6. Return $\hat{Q}_k = f_{\hat{\alpha}_k}$ (*truncation may be needed*)

Return $\pi_K(\cdot) = \arg \max_a \hat{Q}_K(\cdot, a)$ (*greedy policy*)

Linear Fitted Q-iteration: Sampling

1. Draw n samples $(x_i, a_i) \stackrel{\text{i.i.d}}{\sim} \rho$
2. Sample $x'_i \sim p(\cdot | x_i, a_i)$ and $r_i = r(x_i, a_i)$

Linear Fitted Q-iteration: Sampling

1. Draw n samples $(x_i, a_i) \stackrel{\text{i.i.d}}{\sim} \rho$
 2. Sample $x'_i \sim p(\cdot | x_i, a_i)$ and $r_i = r(x_i, a_i)$
- ▶ In practice it can be done *once* before running the algorithm
 - ▶ The sampling distribution ρ should cover the state-action space in all *relevant* regions
 - ▶ If not possible to choose ρ , a *database* of samples can be used

Linear Fitted Q-iteration: The Training Set

4. Compute $y_i = r_i + \gamma \max_a \widehat{Q}_{k-1}(x'_i, a)$
5. Build training set $\{(x_i, a_i), y_i\}_{i=1}^n$

Linear Fitted Q-iteration: The Training Set

4. Compute $y_i = r_i + \gamma \max_a \widehat{Q}_{k-1}(x'_i, a)$
5. Build training set $\{(x_i, a_i), y_i\}_{i=1}^n$

- ▶ Each sample y_i is an unbiased sample, since

$$\begin{aligned}\mathbb{E}[y_i | x_i, a_i] &= \mathbb{E}[r_i + \gamma \max_a \widehat{Q}_{k-1}(x'_i, a)] = r(x_i, a_i) + \gamma \mathbb{E}[\max_a \widehat{Q}_{k-1}(x'_i, a)] \\ &= r(x_i, a_i) + \gamma \int_{\mathcal{X}} \max_a \widehat{Q}_{k-1}(x', a) p(dy | x, a) = \mathcal{T} \widehat{Q}_{k-1}(x_i, a_i)\end{aligned}$$

- ▶ The problem “reduces” to standard *regression*
- ▶ It should be recomputed at each iteration

Linear Fitted Q-iteration: The Regression Problem

6. Solve the *least squares problem*

$$f_{\hat{\alpha}_k} = \arg \min_{f_{\alpha} \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n (f_{\alpha}(x_i, a_i) - y_i)^2$$

7. Return $\hat{Q}_k = f_{\hat{\alpha}_k}$ (*truncation may be needed*)

Linear Fitted Q-iteration: The Regression Problem

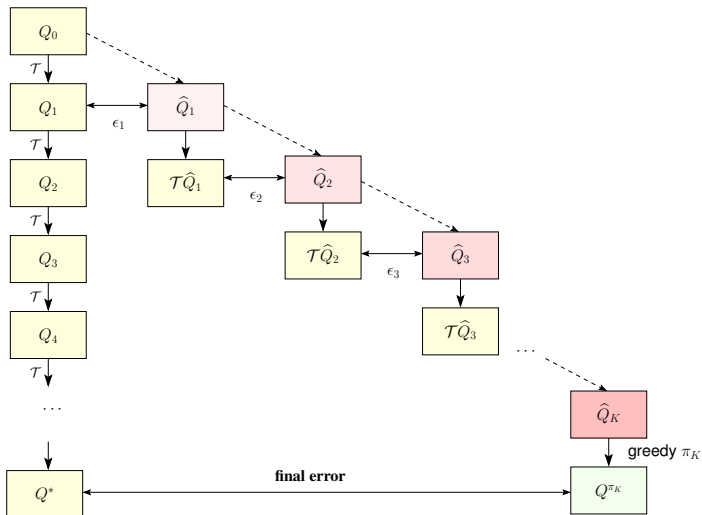
6. Solve the *least squares problem*

$$f_{\hat{\alpha}_k} = \arg \min_{f_{\alpha} \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n (f_{\alpha}(x_i, a_i) - y_i)^2$$

7. Return $\hat{Q}_k = f_{\hat{\alpha}_k}$ (*truncation may be needed*)

- ▶ Thanks to the linear space we can solve it as
 - ▶ Build matrix $\Phi = [\phi(x_1, a_1)^\top \dots \phi(x_n, a_n)^\top]$
 - ▶ Compute $\hat{\alpha}^k = (\Phi^\top \Phi)^{-1} \Phi^\top y$ (*least-squares solution*)
- ▶ Truncation to $[-V_{\max}; V_{\max}]$ (with $V_{\max} = R_{\max}/(1 - \gamma)$)

Sketch of the Analysis



Theoretical Objectives

Objective: derive a bound on the performance (*quadratic*) loss w.r.t. a *testing* distribution μ

$$\|Q^* - Q^{\pi_K}\|_{\mu} \leq ???$$

Theoretical Objectives

Objective: derive a bound on the performance (*quadratic*) loss w.r.t. a *testing* distribution μ

$$\|Q^* - Q^{\pi_K}\|_{\mu} \leq ???$$

Sub-Objective 1: derive an *intermediate* bound on the prediction error at *any* iteration k w.r.t. to the *sampling* distribution ρ

$$\|\mathcal{T}\hat{Q}_{k-1} - \hat{Q}_k\|_{\rho} \leq ???$$

Theoretical Objectives

Objective: derive a bound on the performance (*quadratic*) loss w.r.t. a *testing* distribution μ

$$\|Q^* - Q^{\pi_K}\|_{\mu} \leq ???$$

Sub-Objective 1: derive an *intermediate* bound on the prediction error at *any* iteration k w.r.t. to the *sampling* distribution ρ

$$\|\mathcal{T}\hat{Q}_{k-1} - \hat{Q}_k\|_{\rho} \leq ???$$

Sub-Objective 2: analyze how the *error at each iteration* is *propagated* through iterations

$$\|Q^* - Q^{\pi_K}\|_{\mu} \leq \textit{propagation}(\|\mathcal{T}\hat{Q}_{k-1} - \hat{Q}_k\|_{\rho})$$

The Sources of Error

- ▶ *Desired* solution

$$Q_k = \mathcal{T}\hat{Q}_{k-1}$$

The Sources of Error

- ▶ *Desired* solution

$$Q_k = \mathcal{T}\hat{Q}_{k-1}$$

- ▶ *Best* solution (wrt sampling distribution ρ)

$$f_{\alpha_k^*} = \arg \inf_{f_{\alpha} \in \mathcal{F}} \|f_{\alpha} - Q_k\|_{\rho}$$

The Sources of Error

- ▶ *Desired* solution

$$Q_k = \mathcal{T}\hat{Q}_{k-1}$$

- ▶ *Best* solution (wrt sampling distribution ρ)

$$f_{\alpha_k^*} = \arg \inf_{f_\alpha \in \mathcal{F}} \|f_\alpha - Q_k\|_\rho$$

⇒ **Error** from the approximation space \mathcal{F}

The Sources of Error

- ▶ *Desired* solution

$$Q_k = \mathcal{T}\hat{Q}_{k-1}$$

- ▶ *Best* solution (wrt sampling distribution ρ)

$$f_{\alpha_k^*} = \arg \inf_{f_\alpha \in \mathcal{F}} \|f_\alpha - Q_k\|_\rho$$

\Rightarrow **Error** from the approximation space \mathcal{F}

- ▶ *Returned* solution

$$f_{\hat{\alpha}_k} = \arg \min_{f_\alpha \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n (f_\alpha(x_i, a_i) - y_i)^2$$

The Sources of Error

- ▶ *Desired* solution

$$Q_k = \mathcal{T}\hat{Q}_{k-1}$$

- ▶ *Best* solution (wrt sampling distribution ρ)

$$f_{\alpha_k^*} = \arg \inf_{f_\alpha \in \mathcal{F}} \|f_\alpha - Q_k\|_\rho$$

⇒ **Error** from the approximation space \mathcal{F}

- ▶ *Returned* solution

$$f_{\hat{\alpha}_k} = \arg \min_{f_\alpha \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n (f_\alpha(x_i, a_i) - y_i)^2$$

⇒ **Error** from the (random) samples

Per-Iteration Error

Theorem

At each iteration k , Linear-FQI returns an approximation \widehat{Q}_k such that (**Sub-Objective 1**)

$$\begin{aligned} \|Q_k - \widehat{Q}_k\|_\rho &\leq 4\|Q_k - f_{\alpha_k^*}\|_\rho \\ &\quad + O\left(\left(V_{\max} + L\|\alpha_k^*\|\right)\sqrt{\frac{\log 1/\delta}{n}}\right) \\ &\quad + O\left(V_{\max}\sqrt{\frac{d \log n/\delta}{n}}\right), \end{aligned}$$

with probability $1 - \delta$.

Tools: concentration of measure inequalities, covering space, linear algebra, union bounds, special tricks for linear spaces, ...

Per-Iteration Error

$$\begin{aligned} \|Q_k - \widehat{Q}_k\|_\rho &\leq 4\|Q_k - f_{\alpha_k^*}\|_\rho \\ &+ O\left((V_{\max} + L\|\alpha_k^*\|)\sqrt{\frac{\log 1/\delta}{n}}\right) \\ &+ O\left(V_{\max}\sqrt{\frac{d \log n/\delta}{n}}\right) \end{aligned}$$

Per-Iteration Error

$$\begin{aligned} \|Q_k - \widehat{Q}_k\|_\rho &\leq 4\|Q_k - f_{\alpha_k^*}\|_\rho \\ &\quad + O\left(\left(V_{\max} + L\|\alpha_k^*\|\right)\sqrt{\frac{\log 1/\delta}{n}}\right) \\ &\quad + O\left(V_{\max}\sqrt{\frac{d \log n/\delta}{n}}\right) \end{aligned}$$

Remarks

- ▶ No algorithm can do better
- ▶ Constant **4**
- ▶ Depends on the space \mathcal{F}
- ▶ Changes with the iteration k

Per-Iteration Error

$$\begin{aligned} \|Q_k - \hat{Q}_k\|_\rho &\leq 4\|Q_k - f_{\alpha_k^*}\|_\rho \\ &\quad + O\left((V_{\max} + L\|\alpha_k^*\|)\sqrt{\frac{\log 1/\delta}{n}}\right) \\ &\quad + O\left(V_{\max}\sqrt{\frac{d \log n/\delta}{n}}\right) \end{aligned}$$

Remarks

- ▶ Vanishing to zero as $O(n^{-1/2})$
- ▶ Depends on the features (L) and on the best solution ($\|\alpha_k^*\|$)

Per-Iteration Error

$$\begin{aligned} \|Q_k - \widehat{Q}_k\|_\rho &\leq 4\|Q_k - f_{\alpha_k^*}\|_\rho \\ &+ O\left((V_{\max} + L\|\alpha_k^*\|)\sqrt{\frac{\log 1/\delta}{n}}\right) \\ &+ O\left(V_{\max}\sqrt{\frac{d \log n/\delta}{n}}\right) \end{aligned}$$

Remarks

- ▶ Vanishing to zero as $O(n^{-1/2})$
- ▶ Depends on the dimensionality of the space (d) and the number of samples (n)

Error Propagation

Objective

$$\|Q^* - Q^{\pi_K}\|_{\mu}$$

Error Propagation

Objective

$$\|Q^* - Q^{\pi_K}\|_{\mu}$$

- ▶ **Problem 1:** the test norm μ is different from the sampling norm ρ

Error Propagation

Objective

$$\|Q^* - Q^{\pi_k}\|_{\mu}$$

- ▶ **Problem 1:** the test norm μ is different from the sampling norm ρ
- ▶ **Problem 2:** we have bounds for \hat{Q}_k not for the performance of the corresponding π_k

Error Propagation

Objective

$$\|Q^* - Q^{\pi_K}\|_{\mu}$$

- ▶ **Problem 1:** the test norm μ is different from the sampling norm ρ
- ▶ **Problem 2:** we have bounds for \hat{Q}_k not for the performance of the corresponding π_k
- ▶ **Problem 3:** we have bounds for one single iteration

Error Propagation

Transition kernel for a fixed policy P^π .

- ▶ m -step (worst-case) concentration of future state distribution

$$c(m) = \sup_{\pi_1 \dots \pi_m} \left\| \frac{d(\mu P^{\pi_1} \dots P^{\pi_m})}{d\rho} \right\|_\infty < \infty$$

Error Propagation

Transition kernel for a fixed policy P^π .

- ▶ m -step (worst-case) concentration of future state distribution

$$c(m) = \sup_{\pi_1 \dots \pi_m} \left\| \frac{d(\mu P^{\pi_1} \dots P^{\pi_m})}{d\rho} \right\|_\infty < \infty$$

- ▶ Average (discounted) concentration

$$C_{\mu, \rho} = (1 - \gamma)^2 \sum_{m \geq 1} m \gamma^{m-1} c(m) < +\infty$$

Error Propagation

Remark: relationship to top-Lyapunov exponent

$$L^+ = \sup_{\pi} \limsup_{m \rightarrow \infty} \frac{1}{m} \log^+ (\|\rho P^{\pi_1} P^{\pi_2} \dots P^{\pi_m}\|)$$

If $L^+ \leq 0$ (*stable system*), then $c(m)$ has a growth rate which is polynomial and $C_{\mu, \rho} < \infty$ is *finite*

Error Propagation

Proposition

Let $\epsilon_k = Q_k - \widehat{Q}_k$ be the propagation error at each iteration, then after K iteration the *performance loss* of the greedy policy π_K is

$$\|Q^* - Q^{\pi_K}\|_{\mu}^2 \leq \left[\frac{2\gamma}{(1-\gamma)^2} \right]^2 C_{\mu,\rho} \max_k \|\epsilon_k\|_{\rho}^2 + O\left(\frac{\gamma^K}{(1-\gamma)^3} V_{\max}^2 \right)$$

The Final Bound

Bringing everything together...

$$\|Q^* - Q^{\pi_K}\|_{\mu}^2 \leq \left[\frac{2\gamma}{(1-\gamma)^2} \right]^2 C_{\mu,\rho} \max_k \|\epsilon_k\|_{\rho}^2 + O\left(\frac{\gamma^K}{(1-\gamma)^3} V_{\max}^2 \right)$$

The Final Bound

Bringing everything together...

$$\|Q^* - Q^{\pi_K}\|_{\mu}^2 \leq \left[\frac{2\gamma}{(1-\gamma)^2} \right]^2 C_{\mu,\rho} \max_k \|\epsilon_k\|_{\rho}^2 + O\left(\frac{\gamma^K}{(1-\gamma)^3} V_{\max}^2\right)$$

$$\begin{aligned} \|\epsilon_k\|_{\rho} &= \|Q_k - \hat{Q}_k\|_{\rho} \leq 4\|Q_k - f_{\alpha_k^*}\|_{\rho} \\ &\quad + O\left((V_{\max} + L\|\alpha_k^*\|)\sqrt{\frac{\log 1/\delta}{n}}\right) \\ &\quad + O\left(V_{\max}\sqrt{\frac{d \log n/\delta}{n}}\right) \end{aligned}$$

The Final Bound

Theorem (see e.g., Munos,'03)

LinearFQI with a space \mathcal{F} of d features, with n samples at each iteration returns a policy π_K after K iterations such that

$$\|Q^* - Q^{\pi_K}\|_{\mu} \leq \frac{2\gamma}{(1-\gamma)^2} \sqrt{C_{\mu,\rho}} \left(4d(\mathcal{F}, \mathcal{T}\mathcal{F}) + O\left(V_{\max} \left(1 + \frac{L}{\sqrt{\omega}}\right) \sqrt{\frac{d \log n/\delta}{n}} \right) \right) \\ + O\left(\frac{\gamma^K}{(1-\gamma)^3} V_{\max}^2 \right)$$

The Final Bound

Theorem

LinearFQI with a space \mathcal{F} of d features, with n samples at each iteration returns a policy π_K after K iterations such that

$$\|Q^* - Q^{\pi_K}\|_{\mu} \leq \frac{2\gamma}{(1-\gamma)^2} \sqrt{C_{\mu,\rho}} \left(4d(\mathcal{F}, \mathcal{T}\mathcal{F}) + O\left(V_{\max} \left(1 + \frac{L}{\sqrt{\omega}}\right) \sqrt{\frac{d \log n/\delta}{n}} \right) \right) + O\left(\frac{\gamma^K}{(1-\gamma)^3} V_{\max}^2 \right)$$

The *propagation* (and different norms) makes the problem *more complex*
 \Rightarrow how do we choose the *sampling distribution*?

The Final Bound

Theorem

LinearFQI with a space \mathcal{F} of d features, with n samples at each iteration returns a policy π_K after K iterations such that

$$\|Q^* - Q^{\pi_K}\|_{\mu} \leq \frac{2\gamma}{(1-\gamma)^2} \sqrt{C_{\mu,\rho}} \left(4d(\mathcal{F}, \mathcal{T}\mathcal{F}) + O\left(V_{\max} \left(1 + \frac{L}{\sqrt{\omega}}\right) \sqrt{\frac{d \log n/\delta}{n}} \right) \right) \\ + O\left(\frac{\gamma^K}{(1-\gamma)^3} V_{\max}^2 \right)$$

The *approximation* error is *worse* than in regression

The Final Bound

The inherent Bellman error

$$\begin{aligned}\|Q_k - f_{\alpha_k^*}\|_\rho &= \inf_{f \in \mathcal{F}} \|Q_k - f\|_\rho \\ &= \inf_{f \in \mathcal{F}} \|\mathcal{T}\hat{Q}_{k-1} - f\|_\rho \\ &\leq \inf_{f \in \mathcal{F}} \|\mathcal{T}f_{\alpha_{k-1}} - f\|_\rho \\ &\leq \sup_{g \in \mathcal{F}} \inf_{f \in \mathcal{F}} \|\mathcal{T}g - f\|_\rho = d(\mathcal{F}, \mathcal{TF})\end{aligned}$$

Question: how to design \mathcal{F} to make it “compatible” with the Bellman operator?

The Final Bound

Theorem

LinearFQI with a space \mathcal{F} of d features, with n samples at each iteration returns a policy π_K after K iterations such that

$$\|Q^* - Q^{\pi_K}\|_{\mu} \leq \frac{2\gamma}{(1-\gamma)^2} \sqrt{C_{\mu,\rho}} \left(4d(\mathcal{F}, \mathcal{T}\mathcal{F}) + O\left(V_{\max} \left(1 + \frac{L}{\sqrt{\omega}}\right) \sqrt{\frac{d \log n/\delta}{n}} \right) \right) \\ + O\left(\frac{\gamma^K}{(1-\gamma)^3} V_{\max}^2 \right)$$

The dependency on γ is worse than at each iteration

\Rightarrow is it possible to *avoid* it?

The Final Bound

Theorem

LinearFQI with a space \mathcal{F} of d features, with n samples at each iteration returns a policy π_K after K iterations such that

$$\|Q^* - Q^{\pi_K}\|_{\mu} \leq \frac{2\gamma}{(1-\gamma)^2} \sqrt{C_{\mu,\rho}} \left(4d(\mathcal{F}, \mathcal{T}\mathcal{F}) + O\left(V_{\max} \left(1 + \frac{L}{\sqrt{\omega}}\right) \sqrt{\frac{d \log n/\delta}{n}} \right) \right) \\ + O\left(\frac{\gamma^K}{(1-\gamma)^3} V_{\max}^2 \right)$$

The error decreases exponentially in K

$$\Rightarrow K \approx \epsilon / (1 - \gamma)$$

The Final Bound

Theorem

LinearFQI with a space \mathcal{F} of d features, with n samples at each iteration returns a policy π_K after K iterations such that

$$\|Q^* - Q^{\pi_K}\|_{\mu} \leq \frac{2\gamma}{(1-\gamma)^2} \sqrt{C_{\mu,\rho}} \left(4d(\mathcal{F}, \mathcal{T}\mathcal{F}) + O\left(V_{\max} \left(1 + \frac{L}{\sqrt{\omega}}\right) \sqrt{\frac{d \log n/\delta}{n}} \right) \right) \\ + O\left(\frac{\gamma^K}{(1-\gamma)^3} V_{\max}^2 \right)$$

The smallest eigenvalue of the Gram matrix

\Rightarrow design the features so as to be *orthogonal* w.r.t. ρ

The Final Bound

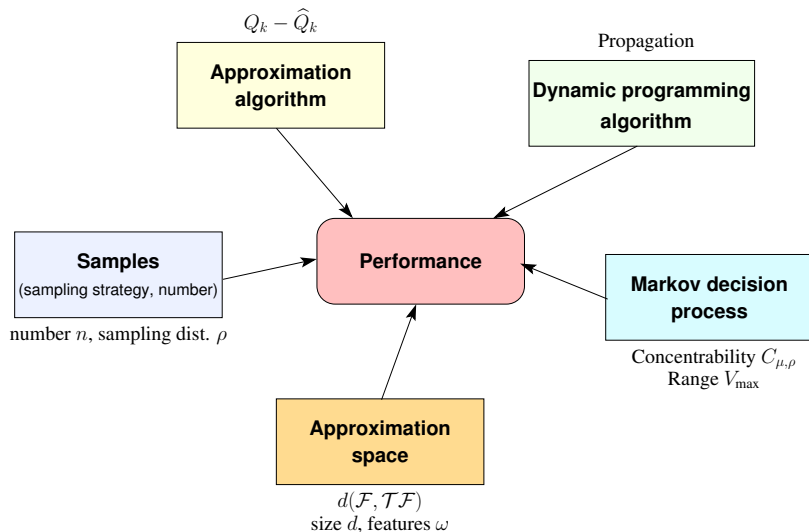
Theorem

LinearFQI with a space \mathcal{F} of d features, with n samples at each iteration returns a policy π_K after K iterations such that

$$\|Q^* - Q^{\pi_K}\|_{\mu} \leq \frac{2\gamma}{(1-\gamma)^2} \sqrt{C_{\mu,\rho}} \left(4d(\mathcal{F}, \mathcal{T}\mathcal{F}) + O\left(V_{\max} \left(1 + \frac{L}{\sqrt{\omega}}\right) \sqrt{\frac{d \log n/\delta}{n}} \right) \right) \\ + O\left(\frac{\gamma^K}{(1-\gamma)^3} V_{\max}^2 \right)$$

The asymptotic rate $O(d/n)$ is the same as for regression

Summary



Other implementations

Replace the *regression* step with

- ▶ K -nearest neighbour
- ▶ Regularized linear regression with L_1 or L_2 regularisation
- ▶ Neural network
- ▶ Support vector regression
- ▶ ...

Example: the Optimal Replacement Problem

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

Example: the Optimal Replacement Problem

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

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

Example: the Optimal Replacement Problem

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

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

Cost:

▶ $c(x, R) = C$

▶ $c(x, K) = c(x)$ maintenance plus extra costs.

Example: the Optimal Replacement Problem

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

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

Cost:

- ▶ $c(x, R) = C$

- ▶ $c(x, K) = c(x)$ maintenance 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 $d(y - x)$.

Example: the Optimal Replacement Problem

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

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

Cost:

- ▶ $c(x, R) = C$

- ▶ $c(x, K) = c(x)$ maintenance 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.

Example: the Optimal Replacement Problem

Optimal value function

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

Example: the Optimal Replacement Problem

Optimal value function

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

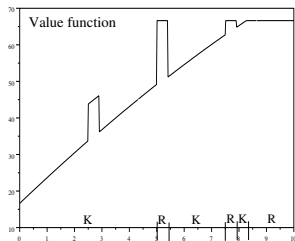
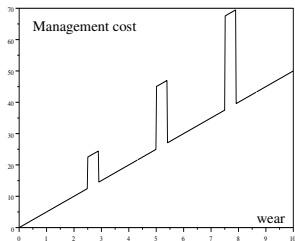
Optimal policy: action that attains the minimum

Example: the Optimal Replacement Problem

Optimal value function

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

Optimal policy: action that attains the minimum

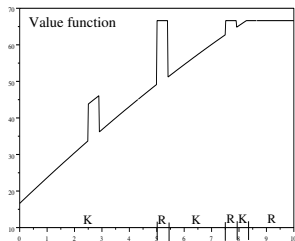
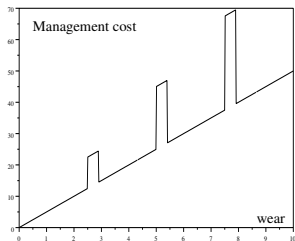


Example: the Optimal Replacement Problem

Optimal value function

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

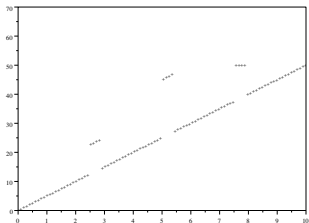
Optimal policy: action that attains the minimum



Linear approximation space $\mathcal{F} := \left\{ V_n(x) = \sum_{k=1}^{20} \alpha_k \cos\left(k\pi \frac{x}{x_{\max}}\right) \right\}$.

Example: the Optimal Replacement Problem

Collect N sample on a uniform grid.



Example: the Optimal Replacement Problem

Collect N sample on a uniform grid.

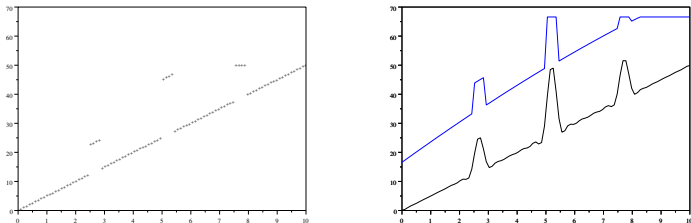


Figure: Left: the *target* values computed as $\{TV_0(x_n)\}_{1 \leq n \leq N}$. Right: the approximation $V_1 \in \mathcal{F}$ of the target function TV_0 .

Example: the Optimal Replacement Problem

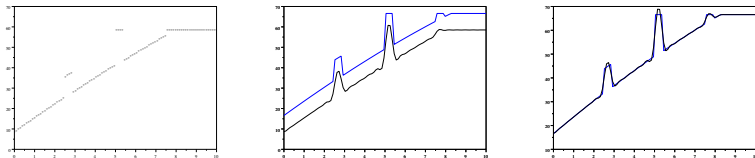


Figure: Left: the *target* values computed as $\{\mathcal{T}V_1(x_n)\}_{1 \leq n \leq N}$. Center: the approximation $V_2 \in \mathcal{F}$ of $\mathcal{T}V_1$. Right: the approximation $V_n \in \mathcal{F}$ after n iterations.

Example: the Optimal Replacement Problem

Simulation

Approximate Dynamic Programming

(a.k.a. Batch Reinforcement Learning)

Approximate Value Iteration

Approximate Policy Iteration

Policy Iteration: the Idea

1. Let π_0 be *any* stationary policy
2. At each iteration $k = 1, 2, \dots, K$
 - ▶ *Policy evaluation* given π_k , compute $V_k = V^{\pi_k}$.
 - ▶ *Policy improvement*: compute the *greedy* policy

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

3. Return the last policy π_K

Policy Iteration: the Idea

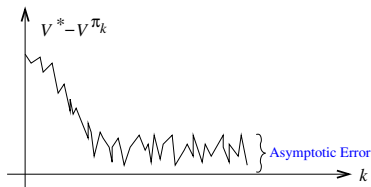
1. Let π_0 be *any* stationary policy
2. At each iteration $k = 1, 2, \dots, K$
 - ▶ *Policy evaluation* given π_k , compute $V_k = V^{\pi_k}$.
 - ▶ *Policy improvement*: compute the *greedy* policy

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

3. Return the last policy π_K
 - ▶ **Problem**: how can we approximate V^{π_k} ?
 - ▶ **Problem**: if $V_k \neq V^{\pi_k}$, does (approx.) policy iteration still work?

Approximate Policy Iteration: performance loss

Problem: the algorithm is no longer guaranteed to converge.



Proposition

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

$$\limsup_{k \rightarrow \infty} \underbrace{\|V^* - V^{\pi_k}\|_{\infty}}_{\text{performance loss}} \leq \frac{2\gamma}{(1-\gamma)^2} \limsup_{k \rightarrow \infty} \underbrace{\|V_k - V^{\pi_k}\|_{\infty}}_{\text{approximation error}}$$

Least-Squares Policy Iteration (LSPI)

LSPI uses

- ▶ Linear space to approximate value functions*

$$\mathcal{F} = \left\{ f(x) = \sum_{j=1}^d \alpha_j \varphi_j(x), \alpha \in \mathbb{R}^d \right\}$$

Least-Squares Policy Iteration (LSPI)

LSPI uses

- ▶ Linear space to approximate value functions*

$$\mathcal{F} = \left\{ f(x) = \sum_{j=1}^d \alpha_j \varphi_j(x), \alpha \in \mathbb{R}^d \right\}$$

- ▶ Least-Squares Temporal Difference (LSTD) algorithm for *policy evaluation*.

*In practice we use approximations of action-value functions.

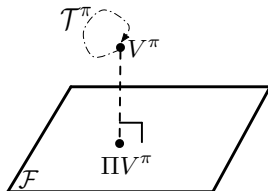
Least-Squares Temporal-Difference Learning (LSTD)

- ▶ V^π may not belong to \mathcal{F}
- ▶ Best approximation of V^π in \mathcal{F} is

$$V^\pi \notin \mathcal{F}$$

$$\Pi V^\pi = \arg \min_{f \in \mathcal{F}} \|V^\pi - f\|$$

(Π is the projection onto \mathcal{F})



Least-Squares Temporal-Difference Learning (LSTD)

- ▶ V^π is the fixed-point of \mathcal{T}^π

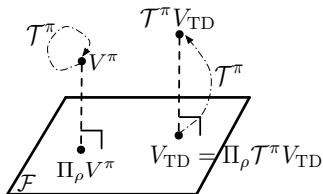
$$V^\pi = \mathcal{T}^\pi V^\pi = r^\pi + \gamma P^\pi V^\pi$$

- ▶ LSTD searches for the fixed-point of $\Pi_{2,\rho} \mathcal{T}^\pi$

$$\Pi_{2,\rho} g = \arg \min_{f \in \mathcal{F}} \|g - f\|_{2,\rho}$$

- ▶ **When** the fixed-point of $\Pi_\rho \mathcal{T}^\pi$ exists, we call it the LSTD solution

$$V_{TD} = \Pi_\rho \mathcal{T}^\pi V_{TD}$$



Least-Squares Temporal-Difference Learning (LSTD)

$$V_{TD} = \Pi_{\rho} \mathcal{T}^{\pi} V_{TD}$$

- ▶ The projection Π_{ρ} is orthogonal *in expectation* w.r.t. the space \mathcal{F} *spanned* by the features $\varphi_1, \dots, \varphi_d$

$$\mathbb{E}_{x \sim \rho} [(\mathcal{T}^{\pi} V_{TD}(x) - V_{TD}(x)) \varphi_i(x)] = 0, \quad \forall i \in [1, d]$$

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

Least-Squares Temporal-Difference Learning (LSTD)

$$V_{TD} = \Pi_{\rho} \mathcal{T}^{\pi} V_{TD}$$

- ▶ The projection Π_{ρ} is orthogonal *in expectation* w.r.t. the space \mathcal{F} *spanned* by the features $\varphi_1, \dots, \varphi_d$

$$\mathbb{E}_{x \sim \rho} [(\mathcal{T}^{\pi} V_{TD}(x) - V_{TD}(x)) \varphi_i(x)] = 0, \quad \forall i \in [1, d]$$

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

- ▶ By definition of Bellman operator

$$\langle r^{\pi} + \gamma P^{\pi} V_{TD} - V_{TD}, \varphi_i \rangle_{\rho} = 0$$

$$\langle r^{\pi}, \varphi_i \rangle_{\rho} - \langle (I - \gamma P^{\pi}) V_{TD}, \varphi_i \rangle_{\rho} = 0$$

Least-Squares Temporal-Difference Learning (LSTD)

$$V_{TD} = \Pi_{\rho} \mathcal{T}^{\pi} V_{TD}$$

- ▶ The projection Π_{ρ} is orthogonal *in expectation* w.r.t. the space \mathcal{F} spanned by the features $\varphi_1, \dots, \varphi_d$

$$\mathbb{E}_{x \sim \rho} [(\mathcal{T}^{\pi} V_{TD}(x) - V_{TD}(x)) \varphi_i(x)] = 0, \quad \forall i \in [1, d]$$

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

- ▶ By definition of Bellman operator

$$\langle r^{\pi} + \gamma P^{\pi} V_{TD} - V_{TD}, \varphi_i \rangle_{\rho} = 0$$

$$\langle r^{\pi}, \varphi_i \rangle_{\rho} - \langle (I - \gamma P^{\pi}) V_{TD}, \varphi_i \rangle_{\rho} = 0$$

- ▶ Since $V_{TD} \in \mathcal{F}$, there exists α_{TD} such that $V_{TD}(x) = \phi(x)^{\top} \alpha_{TD}$

$$\langle r^{\pi}, \varphi_i \rangle_{\rho} - \sum_{j=1}^d \langle (I - \gamma P^{\pi}) \varphi_j \alpha_{TD,j}, \varphi_i \rangle_{\rho} = 0$$

$$\langle r^{\pi}, \varphi_i \rangle_{\rho} - \sum_{j=1}^d \langle (I - \gamma P^{\pi}) \varphi_j, \varphi_i \rangle_{\rho} \alpha_{TD,j} = 0$$

Least-Squares Temporal-Difference Learning (LSTD)

$$V_{TD} = \Pi_{\rho} \mathcal{T}^{\pi} V_{TD}$$

⇓

$$\underbrace{\langle r^{\pi}, \varphi_i \rangle_{\rho}}_{b_i} - \sum_{j=1}^d \underbrace{\langle (I - \gamma P^{\pi}) \varphi_j, \varphi_i \rangle_{\rho}}_{A_{i,j}} \alpha_{TD,j} = 0$$

⇓

$$A \alpha_{TD} = b$$

Least-Squares Temporal-Difference Learning (LSTD)

- ▶ **Problem:** In general, $\Pi_{\rho} \mathcal{T}^{\pi}$ is **not a contraction** and does not have a fixed-point.
- ▶ **Solution:** If $\rho = \rho^{\pi}$ (*stationary dist. of π*) then $\Pi_{\rho^{\pi}} \mathcal{T}^{\pi}$ has a unique fixed-point.

Least-Squares Temporal-Difference Learning (LSTD)

- ▶ **Problem:** In general, $\Pi_{\rho} \mathcal{T}^{\pi}$ is **not a contraction** and does not have a fixed-point.
- ▶ **Solution:** If $\rho = \rho^{\pi}$ (*stationary dist. of π*) then $\Pi_{\rho^{\pi}} \mathcal{T}^{\pi}$ has a unique fixed-point.
- ▶ **Problem:** In general, $\Pi_{\rho} \mathcal{T}^{\pi}$ cannot be computed (because **unknown**)
- ▶ **Solution:** Use *samples* coming from a “trajectory” of π .

Least-Squares Policy Iteration (LSPI)

Input: space \mathcal{F} , iterations K , sampling distribution ρ , num of samples n

Least-Squares Policy Iteration (LSPI)

Input: space \mathcal{F} , iterations K , sampling distribution ρ , num of samples n

Initial policy π_0

Least-Squares Policy Iteration (LSPI)

Input: space \mathcal{F} , iterations K , sampling distribution ρ , num of samples n

Initial policy π_0

For $k = 1, \dots, K$

Least-Squares Policy Iteration (LSPI)

Input: space \mathcal{F} , iterations K , sampling distribution ρ , num of samples n

Initial policy π_0

For $k = 1, \dots, K$

1. Generate a trajectory of length n from the stationary dist. ρ^{π_k}
 $(x_1, \pi_k(x_1), r_1, x_2, \pi_k(x_2), r_2, \dots, x_{n-1}, \pi_k(x_{n-1}), r_{n-1}, x_n)$

Least-Squares Policy Iteration (LSPI)

Input: space \mathcal{F} , iterations K , sampling distribution ρ , num of samples n

Initial policy π_0

For $k = 1, \dots, K$

1. Generate a trajectory of length n from the stationary dist. ρ^{π_k}
 $(x_1, \pi_k(x_1), r_1, x_2, \pi_k(x_2), r_2, \dots, x_{n-1}, \pi_k(x_{n-1}), r_{n-1}, x_n)$

2. Compute the empirical matrix \hat{A}_k and the vector \hat{b}_k

$$[\hat{A}_k]_{i,j} = \frac{1}{n} \sum_{t=1}^n (\varphi_j(x_t) - \gamma \varphi_j(x_{t+1})) \varphi_i(x_t) \approx \langle (I - \gamma P^\pi) \varphi_j, \varphi_i \rangle_{\rho^{\pi_k}}$$

$$[\hat{b}_k]_i = \frac{1}{n} \sum_{t=1}^n \varphi_i(x_t) r_t \approx \langle r^\pi, \varphi_i \rangle_{\rho^{\pi_k}}$$

3. Solve the linear system $\alpha_k = \hat{A}_k^{-1} \hat{b}_k$

Least-Squares Policy Iteration (LSPI)

Input: space \mathcal{F} , iterations K , sampling distribution ρ , num of samples n

Initial policy π_0

For $k = 1, \dots, K$

1. Generate a trajectory of length n from the stationary dist. ρ^{π_k}
 $(x_1, \pi_k(x_1), r_1, x_2, \pi_k(x_2), r_2, \dots, x_{n-1}, \pi_k(x_{n-1}), r_{n-1}, x_n)$

2. Compute the empirical matrix \hat{A}_k and the vector \hat{b}_k

$$[\hat{A}_k]_{i,j} = \frac{1}{n} \sum_{t=1}^n (\varphi_j(x_t) - \gamma \varphi_j(x_{t+1})) \varphi_i(x_t) \approx \langle (I - \gamma P^\pi) \varphi_j, \varphi_i \rangle_{\rho^{\pi_k}}$$

$$[\hat{b}_k]_i = \frac{1}{n} \sum_{t=1}^n \varphi_i(x_t) r_t \approx \langle r^\pi, \varphi_i \rangle_{\rho^{\pi_k}}$$

3. Solve the linear system $\alpha_k = \hat{A}_k^{-1} \hat{b}_k$
4. Compute the greedy policy π_{k+1} w.r.t. $\hat{V}_k = f_{\alpha_k}$

Least-Squares Policy Iteration (LSPI)

Input: space \mathcal{F} , iterations K , sampling distribution ρ , num of samples n

Initial policy π_0

For $k = 1, \dots, K$

1. Generate a trajectory of length n from the stationary dist. ρ^{π_k}
 $(x_1, \pi_k(x_1), r_1, x_2, \pi_k(x_2), r_2, \dots, x_{n-1}, \pi_k(x_{n-1}), r_{n-1}, x_n)$
2. Compute the empirical matrix \hat{A}_k and the vector \hat{b}_k

$$[\hat{A}_k]_{i,j} = \frac{1}{n} \sum_{t=1}^n (\varphi_j(x_t) - \gamma \varphi_j(x_{t+1})) \varphi_i(x_t) \approx \langle (I - \gamma P^\pi) \varphi_j, \varphi_i \rangle_{\rho^{\pi_k}}$$

$$[\hat{b}_k]_i = \frac{1}{n} \sum_{t=1}^n \varphi_i(x_t) r_t \approx \langle r^\pi, \varphi_i \rangle_{\rho^{\pi_k}}$$

3. Solve the linear system $\alpha_k = \hat{A}_k^{-1} \hat{b}_k$
4. Compute the greedy policy π_{k+1} w.r.t. $\hat{V}_k = f_{\alpha_k}$

Return the last policy π_K

Least-Squares Policy Iteration (LSPI)

1. Generate a trajectory of length n from the stationary dist. ρ^{π_k}
 $(x_1, \pi_k(x_1), r_1, x_2, \pi_k(x_2), r_2, \dots, x_{n-1}, \pi_k(x_{n-1}), r_{n-1}, x_n)$

- ▶ The first few samples may be *discarded* because not actually drawn from the *stationary* distribution ρ^{π_k}
- ▶ *Off-policy* samples could be used with *importance weighting*
- ▶ In practice i.i.d. states drawn from an arbitrary distribution (but with actions π_k) may be used

Least-Squares Policy Iteration (LSPI)

4. Compute the greedy policy π_{k+1} w.r.t. $\hat{V}_k = f_{\alpha_k}$
 - ▶ Computing the greedy policy from \hat{V}_k is difficult, so move to LSTD-Q and compute

$$\pi_{k+1}(x) = \arg \max_a \hat{Q}_k(x, a)$$

Least-Squares Policy Iteration (LSPI)

For $k = 1, \dots, K$

Least-Squares Policy Iteration (LSPI)

For $k = 1, \dots, K$

1. Generate a trajectory of length n from the stationary dist. ρ^{π_k}
 $(x_1, \pi_k(x_1), r_1, x_2, \pi_k(x_2), r_2, \dots, x_{n-1}, \pi_k(x_{n-1}), r_{n-1}, x_n)$

...

4. Compute the greedy policy π_{k+1} w.r.t. $\hat{V}_k = f_{\alpha_k}$

Problem: This process may be unstable because π_k *does not cover* the state space *properly*

▶ Skip Theory

LSTD Algorithm

When $n \rightarrow \infty$ then $\hat{A} \rightarrow A$ and $\hat{b} \rightarrow b$, and thus,

$$\hat{\alpha}_{\text{TD}} \rightarrow \alpha_{\text{TD}} \text{ and } \hat{V}_{\text{TD}} \rightarrow V_{\text{TD}}$$

Proposition (LSTD Performance)

If LSTD is used to estimate the value of π with an **infinite** number of samples drawn from the stationary distribution ρ^π then

$$\|V^\pi - V_{\text{TD}}\|_{\rho^\pi} \leq \frac{1}{\sqrt{1-\gamma^2}} \inf_{V \in \mathcal{F}} \|V^\pi - V\|_{\rho^\pi}$$

LSTD Algorithm

When $n \rightarrow \infty$ then $\hat{A} \rightarrow A$ and $\hat{b} \rightarrow b$, and thus,

$$\hat{\alpha}_{\text{TD}} \rightarrow \alpha_{\text{TD}} \text{ and } \hat{V}_{\text{TD}} \rightarrow V_{\text{TD}}$$

Proposition (LSTD Performance)

If LSTD is used to estimate the value of π with an **infinite** number of samples drawn from the stationary distribution ρ^π then

$$\|V^\pi - V_{\text{TD}}\|_{\rho^\pi} \leq \frac{1}{\sqrt{1-\gamma^2}} \inf_{V \in \mathcal{F}} \|V^\pi - V\|_{\rho^\pi}$$

Problem: we don't have an infinite number of samples...

LSTD Algorithm

When $n \rightarrow \infty$ then $\hat{A} \rightarrow A$ and $\hat{b} \rightarrow b$, and thus,

$$\hat{\alpha}_{\text{TD}} \rightarrow \alpha_{\text{TD}} \text{ and } \hat{V}_{\text{TD}} \rightarrow V_{\text{TD}}$$

Proposition (LSTD Performance)

If LSTD is used to estimate the value of π with an **infinite** number of samples drawn from the stationary distribution ρ^π then

$$\|V^\pi - V_{\text{TD}}\|_{\rho^\pi} \leq \frac{1}{\sqrt{1 - \gamma^2}} \inf_{V \in \mathcal{F}} \|V^\pi - V\|_{\rho^\pi}$$

Problem: we don't have an infinite number of samples...

Problem 2: V_{TD} is a fixed point solution and not a standard machine learning problem...

LSTD Error Bound

Assumption: The Markov chain induced by the policy π_k has a stationary distribution ρ^{π_k} and it is ergodic and β -mixing.

LSTD Error Bound

Assumption: The Markov chain induced by the policy π_k has a stationary distribution ρ^{π_k} and it is ergodic and β -mixing.

Theorem (LSTD Error Bound)

At any iteration k , if LSTD uses n samples obtained from a single trajectory of π and a d -dimensional space, then with probability $1 - \delta$

$$\|V^{\pi_k} - \widehat{V}_k\|_{\rho^{\pi_k}} \leq \frac{c}{\sqrt{1-\gamma^2}} \inf_{f \in \mathcal{F}} \|V^{\pi_k} - f\|_{\rho^{\pi_k}} + O\left(\sqrt{\frac{d \log(d/\delta)}{n \nu}}\right)$$

LSTD Error Bound

$$\|V^\pi - \widehat{V}\|_{\rho^\pi} \leq \frac{c}{\sqrt{1-\gamma^2}} \underbrace{\inf_{f \in \mathcal{F}} \|V^\pi - f\|_{\rho^\pi}}_{\text{approximation error}} + \underbrace{O\left(\sqrt{\frac{d \log(d/\delta)}{n \nu}}\right)}_{\text{estimation error}}$$

- ▶ **Approximation error:** it depends on how well the function space \mathcal{F} can approximate the value function V^π
- ▶ **Estimation error:** it depends on the number of samples n , the dim of the function space d , the smallest eigenvalue of the Gram matrix ν , the mixing properties of the Markov chain (hidden in O)

LSTD Error Bound

$$\|V^{\pi_k} - \widehat{V}_k\|_{\rho^{\pi_k}} \leq \frac{c}{\sqrt{1-\gamma^2}} \underbrace{\inf_{f \in \mathcal{F}} \|V^{\pi_k} - f\|_{\rho^{\pi_k}}}_{\text{approximation error}} + \underbrace{O\left(\sqrt{\frac{d \log(d/\delta)}{n \nu_k}}\right)}_{\text{estimation error}}$$

- ▶ n number of samples and d dimensionality

LSTD Error Bound

$$\|V^{\pi_k} - \widehat{V}_k\|_{\rho^{\pi_k}} \leq \frac{c}{\sqrt{1-\gamma^2}} \underbrace{\inf_{f \in \mathcal{F}} \|V^{\pi_k} - f\|_{\rho^{\pi_k}}}_{\text{approximation error}} + \underbrace{O\left(\sqrt{\frac{d \log(d/\delta)}{n \nu_k}}\right)}_{\text{estimation error}}$$

- ▶ ν_k = the smallest eigenvalue of the Gram matrix $(\int \varphi_i \varphi_j d\rho^{\pi_k})_{i,j}$
(**Assumption:** eigenvalues of the Gram matrix are strictly positive - existence of the model-based LSTD solution)
- ▶ β -mixing coefficients are hidden in the $O(\cdot)$ notation

LSPI Error Bound

Theorem (LSPI Error Bound)

If LSPI is run over K iterations, then the performance loss policy π_K is

$$\|V^* - V^{\pi_K}\|_{\mu} \leq \frac{4\gamma}{(1-\gamma)^2} \left\{ \sqrt{CC_{\mu,\rho}} \left[E_0(\mathcal{F}) + O\left(\sqrt{\frac{d \log(dK/\delta)}{n \nu_{\rho}}}\right) \right] + \gamma^K R_{\max} \right\}$$

with probability $1 - \delta$.

LSPI Error Bound

Theorem (LSPI Error Bound)

If LSPI is run over K iterations, then the performance loss policy π_K is

$$\|V^* - V^{\pi_K}\|_{\mu} \leq \frac{4\gamma}{(1-\gamma)^2} \left\{ \sqrt{CC_{\mu,\rho}} \left[cE_0(\mathcal{F}) + O\left(\sqrt{\frac{d \log(dK/\delta)}{n \nu_{\rho}}}\right) \right] + \gamma^K R_{\max} \right\}$$

with probability $1 - \delta$.

- ▶ **Approximation error:** $E_0(\mathcal{F}) = \sup_{\pi \in \mathcal{G}(\tilde{\mathcal{F}})} \inf_{f \in \mathcal{F}} \|V^{\pi} - f\|_{\rho^{\pi}}$

LSPI Error Bound

Theorem (LSPI Error Bound)

If LSPI is run over K iterations, then the performance loss policy π_K is

$$\|V^* - V^{\pi_K}\|_{\mu} \leq \frac{4\gamma}{(1-\gamma)^2} \left\{ \sqrt{CC_{\mu,\rho}} \left[cE_0(\mathcal{F}) + \mathcal{O} \left(\sqrt{\frac{d \log(dK/\delta)}{n \nu_{\rho}}} \right) \right] + \gamma^K R_{\max} \right\}$$

with probability $1 - \delta$.

- ▶ **Approximation error:** $E_0(\mathcal{F}) = \sup_{\pi \in \mathcal{G}(\tilde{\mathcal{F}})} \inf_{f \in \mathcal{F}} \|V^{\pi} - f\|_{\rho^{\pi}}$
- ▶ **Estimation error:** depends on n, d, ν_{ρ}, K

LSPI Error Bound

Theorem (LSPI Error Bound)

If LSPI is run over K iterations, then the performance loss policy π_K is

$$\|V^* - V^{\pi_K}\|_{\mu} \leq \frac{4\gamma}{(1-\gamma)^2} \left\{ \sqrt{CC_{\mu,\rho}} \left[cE_0(\mathcal{F}) + O\left(\sqrt{\frac{d \log(dK/\delta)}{n \nu_{\rho}}}\right) \right] + \gamma^K R_{\max} \right\}$$

with probability $1 - \delta$.

- ▶ **Approximation error:** $E_0(\mathcal{F}) = \sup_{\pi \in \mathcal{G}(\tilde{\mathcal{F}})} \inf_{f \in \mathcal{F}} \|V^{\pi} - f\|_{\rho^{\pi}}$
- ▶ **Estimation error:** depends on n, d, ν_{ρ}, K
- ▶ **Initialization error:** error due to the choice of the initial value function or initial policy $|V^* - V^{\pi_0}|$

LSPI Error Bound

LSPI Error Bound

$$\|V^* - V^{\pi_K}\|_{\mu} \leq \frac{4\gamma}{(1-\gamma)^2} \left\{ \sqrt{CC_{\mu,\rho}} \left[cE_0(\mathcal{F}) + O\left(\sqrt{\frac{d \log(dK/\delta)}{n \nu_{\rho}}}\right) \right] + \gamma^K R_{\max} \right\}$$

Lower-Bounding Distribution

There exists a distribution ρ such that for any policy $\pi \in \mathcal{G}(\tilde{\mathcal{F}})$, we have $\rho \leq C\rho^{\pi}$, where $C < \infty$ is a constant and ρ^{π} is the stationary distribution of π . Furthermore, we can define the **concentrability** coefficient $C_{\mu,\rho}$ as before.

LSPI Error Bound

LSPI Error Bound

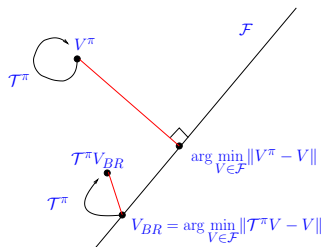
$$\|V^* - V^{\pi_K}\|_{\mu} \leq \frac{4\gamma}{(1-\gamma)^2} \left\{ \sqrt{C C_{\mu,\rho}} \left[cE_0(\mathcal{F}) + O\left(\sqrt{\frac{d \log(dK/\delta)}{n \nu_{\rho}}}\right) \right] + \gamma^K R_{\max} \right\}$$

Lower-Bounding Distribution

There exists a distribution ρ such that for any policy $\pi \in \mathcal{G}(\tilde{\mathcal{F}})$, we have $\rho \leq C\rho^{\pi}$, where $C < \infty$ is a constant and ρ^{π} is the stationary distribution of π . Furthermore, we can define the **concentrability** coefficient $C_{\mu,\rho}$ as before.

- ▶ ν_{ρ} = the smallest eigenvalue of the Gram matrix $(\int \varphi_i \varphi_j d\rho)_{i,j}$

Bellman Residual Minimization (BRM): the idea



Let μ be a distribution over X , V_{BR} is the minimum *Bellman residual w.r.t. T^π*

$$V_{BR} = \arg \min_{V \in \mathcal{F}} \|T^\pi V - V\|_{2, \mu}$$

Bellman Residual Minimization (BRM): the idea

The mapping $\alpha \rightarrow \mathcal{T}^\pi V_\alpha - V_\alpha$ is affine

The function $\alpha \rightarrow \|\mathcal{T}^\pi V_\alpha - V_\alpha\|_\mu^2$ is quadratic

\Rightarrow The minimum is obtained by computing the *gradient and setting it to zero*

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

which can be rewritten as $A\alpha = b$, with

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

Bellman Residual Minimization (BRM): the idea

Remark: the system admits a solution whenever the features ϕ_i are *linearly independent* w.r.t. μ

Bellman Residual Minimization (BRM): the idea

Remark: the system admits a solution whenever the features ϕ_i are *linearly independent* w.r.t. μ

Remark: let $\{\psi_i = \phi_i - \gamma P^\pi \phi_i\}_{i=1\dots d}$, then the previous system can be interpreted as a linear regression problem

$$\|\alpha \cdot \psi - r^\pi\|_\mu$$

BRM: the approximation error

Proposition

We have

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

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

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

BRM: the implementation

Assumption. A generative model is available.

- ▶ Drawn n states $X_t \sim \mu$
- ▶ Call generative model on (X_t, A_t) (with $A_t = \pi(X_t)$) and obtain $R_t = r(X_t, A_t)$, $Y_t \sim p(\cdot | X_t, A_t)$
- ▶ Compute

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

BRM: the implementation

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

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

\Rightarrow minimizing $\hat{\mathcal{B}}(V)$ *does not* correspond to minimizing $\mathcal{B}(V)$ (even when $n \rightarrow \infty$).

BRM: the implementation

Solution. In each state X_t , generate *two independent samples* Y_t
et $Y'_t \sim p(\cdot | X_t, A_t)$

Define

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

$\Rightarrow \hat{\mathcal{B}} \rightarrow \mathcal{B}$ for $n \rightarrow \infty$.

BRM: the implementation

The function $\alpha \rightarrow \hat{\mathcal{B}}(V_\alpha)$ is quadratic and we obtain the linear system

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

BRM: the approximation error

Proof. We 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

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

and

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

BRM: the approximation error

Proof. If we consider the stationary distribution μ_π , then $\|P^\pi\|_{\mu_\pi} = 1$. The matrix $(I - \gamma P^\pi)$ can be written as the power series $\sum_t \gamma (P^\pi)^t$. Applying the norm we obtain

$$\|(I - \gamma P^\pi)^{-1}\|_{\mu_\pi} \leq \sum_{t \geq 0} \gamma^t \|P^\pi\|_{\mu_\pi}^t \leq \frac{1}{1 - \gamma}$$



LSTD vs BRM

- ▶ **Different assumptions:** BRM requires a *generative model*, LSTD requires a *single trajectory*.
- ▶ **The performance is evaluated differently:** BRM *any* distribution, LSTD *stationary* distribution μ^π .

Bibliography I

Reinforcement Learning

The Inria logo is displayed in a white rounded square with a teal border. The word "Inria" is written in a red, cursive script font.

Inria

Alessandro Lazaric

alessandro.lazaric@inria.fr

sequel.lille.inria.fr