Approximate Dynamic Programming

A. LAZARIC (SequeL Team @INRIA-Lille)
ENS Cachan - Master 2 MVA
Value Iteration: the Idea

1. Let $V_0$ be any vector in $\mathbb{R}^N$
2. At each iteration $k = 1, 2, \ldots, K$
   - Compute $V_{k+1} = TV_k$
3. Return the greedy policy

$$
\pi_K(x) \in \arg \max_{a \in A} \left[ r(x, a) + \gamma \sum_y p(y|x, a)V_K(y) \right].
$$
Value Iteration: the Guarantees

From the *fixed point* property of $\mathcal{T}$:

$$\lim_{k \to \infty} V_k = V^*$$

From the *contraction* property of $\mathcal{T}$

$$\|V_{k+1} - V^*\|_\infty \leq \gamma^{k+1} \|V_0 - V^*\|_\infty \to 0$$

*Problem*: what if $V_{k+1} \neq \mathcal{T} V_k$?
Policy Iteration: the Idea

1. Let $\pi_0$ be any stationary policy
2. At each iteration $k = 1, 2, \ldots, K$
   - Policy evaluation given $\pi_k$, compute $V_k = V^{\pi_k}$.
   - Policy improvement: compute the greedy policy
     \[
     \pi_{k+1}(x) \in \arg\max_{a \in A} \left[ r(x, a) + \gamma \sum_y p(y|x, a) V^{\pi_k}(y) \right].
     \]
3. Return the last policy $\pi_K$
Policy Iteration: the Guarantees

The policy iteration algorithm generates a sequence of policies with non-decreasing performance

\[ V^{\pi_{k+1}} \geq V^{\pi_k}, \]

and it converges to \( \pi^* \) in a finite number of iterations.

**Problem**: what if \( V_k \neq V^{\pi_k} \)?
Sources of Error

- **Approximation error.** If $X$ is *large* or *continuous*, value functions $V$ cannot be *represented* correctly
  ⇒ use an *approximation space* $\mathcal{F}$

- **Estimation error.** If the reward $r$ and dynamics $p$ are *unknown*, the Bellman operators $\mathcal{T}$ and $\mathcal{T}^\pi$ cannot be *computed* exactly
  ⇒ *estimate* the Bellman operators from *samples*
In This Lecture

- Infinite horizon setting with discount $\gamma$
- Study the impact of approximation error
- Study the impact of estimation error in the next lecture
Outline

Performance Loss

Approximate Value Iteration

Approximate Policy Iteration
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*

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

i.e.

**performance loss** $= \| V^* - V^\pi \|$
**Proposition**

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

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

*performance loss* \( \leq \) *approx. error*

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

\[ \| V^* - V^\pi \|_\infty \leq \| TV^* - T^\pi V \|_\infty + \| T^\pi V - T^\pi V^\pi \|_\infty \]
\[ \leq \| TV^* - TV \|_\infty + \gamma \| V - V^\pi \|_\infty \]
\[ \leq \gamma \| V^* - V \|_\infty + \gamma (\| V - V^* \|_\infty + \| V^* - V^\pi \|_\infty) \]
\[ \leq \frac{2\gamma}{1 - \gamma} \| V^* - V \|_\infty. \]
From Approximation Error to Performance Loss

**Question:** how do we compute $V$?

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

**Objective:** given an *approximation space* $\mathcal{F}$, compute an approximation $V$ which is as close as possible to the *best approximation* of $V^*$ in $\mathcal{F}$, i.e.

$$V \approx \arg \inf_{f \in \mathcal{F}} ||V^* - f||$$
Outline

Performance Loss

Approximate Value Iteration

Approximate Policy Iteration
Approximate Value Iteration: the Idea

Let $A$ be an \textit{approximation operator}.

1. Let $V_0$ be \textit{any} vector in $\mathbb{R}^N$
2. At each iteration $k = 1, 2, \ldots, K$
   - Compute $V_{k+1} = AV_k$
3. Return the \textit{greedy} policy

$$\pi_K(x) \in \arg \max_{a \in A} \left[ r(x, a) + \gamma \sum_y p(y|x, a) V_K(y) \right].$$
Approximate Value Iteration: the Idea

Let \( A = \Pi_\infty \) be a projection operator in \( L_\infty \)-norm, which corresponds to

\[
V_{k+1} = \Pi_\infty \mathcal{T} V_k = \arg \inf_{V \in \mathcal{F}} \| \mathcal{T} V_k - V \|_\infty
\]
Approximate Value Iteration: convergence

**Proposition**

The projection $\Pi_\infty$ is a *non-expansion* and the joint operator $\Pi_\infty \mathcal{T}$ is a *contraction*.

Then there exists a unique fixed point $\tilde{V} = \Pi_\infty \mathcal{T} \tilde{V}$ which guarantees the *convergence* of AVI.
Approximate Value Iteration: performance loss

Proposition (Bertsekas & Tsitsiklis, 1996)

Let $V^K$ be the function returned by AVI after $K$ iterations and $\pi_K$ its corresponding greedy policy. Then

$$\| V^* - V^\pi_K \|_\infty \leq \frac{2\gamma}{(1 - \gamma)^2} \max_{0 \leq k < K} \| TV_k - ATV_k \|_\infty + \frac{2\gamma^{K+1}}{1 - \gamma} \| V^* - V_0 \|_\infty.$$
Approximate Value Iteration: performance loss

**Proof.** Let $\varepsilon = \max_{0 \leq k < K} \|TV_k - ATV_k\|_{\infty}$.

For any $0 \leq k < K$ we have

$$
\|V^* - V_{k+1}\|_{\infty} \leq \|TV^* - TV_k\|_{\infty} + \|TV_k - V_{k+1}\|_{\infty}
\leq \gamma \|V^* - V_k\|_{\infty} + \varepsilon,
$$

then

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

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

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

Assumption: access to a generative model.

Idea: work with $Q$-functions and linear spaces.

$Q^*$ is the unique fixed point of $\mathcal{T}$ defined over $X \times A$ as:

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

$\mathcal{F}$ is a space defined by $d$ features $\phi_1, \ldots, \phi_d : X \times A \to \mathbb{R}$ as:

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

⇒ At each iteration compute $Q_{k+1} = \Pi_\infty \mathcal{T} Q_k$
Fitted Q-iteration with linear approximation

⇒ At each iteration compute $Q_{k+1} = \Pi_{\infty} \mathcal{T} Q_k$

**Problems:**

- the $\Pi_{\infty}$ operator cannot be computed *efficiently*
- the Bellman operator $\mathcal{T}$ is often *unknown*
Fitted Q-iteration with linear approximation

**Problem**: the $\Pi_\infty$ operator cannot be computed efficiently.

Let $\mu$ a distribution over $X$. We use a projection in $L_{2,\mu}$-norm onto the space $\mathcal{F}$:

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

**Problem:** the Bellman operator $T$ is often *unknown.*

1. Sample $n$ state actions $(X_i, A_i)$ with $X_i \sim \mu$ and $A_i$ random,
2. Simulate $Y_i \sim p(\cdot|X_i, A_i)$ and $R_i = r(X_i, A_i, Y_i)$ with the generative model,
3. Estimate $TQ_k(X_i, A_i)$ with

$$Z_i = R_i + \gamma \max_{a \in A} Q_k(Y_i, a)$$

(unbiased $\mathbb{E}[Z_i|X_i, A_i] = TQ_k(X_i, A_i)$).
Fitted Q-iteration with linear approximation

At each iteration $k$ compute $Q_{k+1}$ as

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

⇒ Since $Q_\alpha$ is a linear function in $\alpha$, the problem is a simple quadratic minimization problem with closed form solution.
Other implementations

- $K$-nearest neighbour
- Regularized linear regression with $L_1$ or $L_2$ regularisation
- Neural network
- Support vector machine
Example: the Optimal Replacement Problem

State: level of wear of an object (e.g., a car).
Action: \{(R)eplace, (K)eeep\}.
Cost:
  - $c(x, R) = C$
  - $c(x, K) = c(x)$ maintenance plus extra costs.

Dynamics:
  - $p(·|x, R) = \exp(\beta)$ with density $d(y) = \beta \exp^{-\beta y} \mathbb{1}\{y \geq 0\}$,
  - $p(·|x, K) = x + \exp(\beta)$ with density $d(y - x)$.

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

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** \( \mathcal{F} := \left\{ V_n(x) = \sum_{k=1}^{20} \alpha_k \cos(k\pi \frac{x}{x_{\text{max}}}) \right\} \).
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.
Outline

Performance Loss

Approximate Value Iteration

Approximate Policy Iteration
  - Linear Temporal-Difference
  - Least-Squares Temporal Difference
  - Bellman Residual Minimization
Approximate Policy Iteration: the Idea

Let $\mathcal{A}$ be an approximation operator.

- **Policy evaluation:** given the current policy $\pi_k$, compute $V_k = \mathcal{A}V^\pi_k$
- **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:** the algorithm is no longer guaranteed to converge.
Proposition

The asymptotic performance of the policies $\pi_k$ 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

- Approximation error: \( e_k = V_k - V^{\pi_k} \),
- Performance gain: \( g_k = V^{\pi_{k+1}} - V^{\pi_k} \),
- Performance loss: \( l_k = V^* - V^{\pi_k} \).
Approximate Policy Iteration: performance loss

Proof (cont’d).
Since $\pi_{k+1}$ is greedy w.r.t. $V_k$ we have that $T^{\pi_{k+1}} V_k \geq T^{\pi_k} V_k$.

$$g_k = T^{\pi_{k+1}} V^{\pi_{k+1}} - T^{\pi_{k+1}} V^{\pi_k} + T^{\pi_{k+1}} V^{\pi_k} - T^{\pi_{k+1}} V_k + T^{\pi_{k+1}} V_k - T^{\pi_k} V_k + T^{\pi_k} V_k - T^{\pi_k} V^{\pi_k}$$

\[(a) \geq \gamma P^{\pi_{k+1}} g_k - \gamma (P^{\pi_{k+1}} - P^{\pi_k}) e_k\]

\[(b) \geq -\gamma (I - \gamma P^{\pi_{k+1}})^{-1} (P^{\pi_{k+1}} - P^{\pi_k}) e_k\]

Which leads to

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

(1)
Approximate Policy Iteration: performance loss

Proof (cont’d).
Relationship between the performance at subsequent iterations. Since \( T^\pi* V_k \leq T^\pi_{k+1} V_k \) we have

\[
l_{k+1} = T^\pi* V^* - T^\pi* V^\pi_k + T^\pi* V^\pi_k - T^\pi* V_k + T^\pi* V_k - T^\pi_{k+1} V_k + T^\pi_{k+1} V_{k} - T^\pi_{k+1} V^\pi_k + T^\pi_{k+1} V^\pi_k - T^\pi_{k+1} V^\pi_{k+1}
\]

\[
\leq \gamma [P^\pi* l_k - P^\pi_{k+1} g_k + (P^\pi_{k+1} - P^\pi*) e_k] .
\]

If we now plug-in equation (1),

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

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

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

\[
l_{k+1} \leq \gamma P^\pi* l_k + \gamma [P^\pi_{k+1} (I - \gamma P^\pi_{k+1})^{-1} (I - \gamma P^\pi_k) - P^\pi*] e_k .
\]
Approximate Policy Iteration: performance loss

Proof (cont’d).
Move to 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,

\[
\limsup_{k \to \infty} (I - \gamma P^\pi^*) \leq \limsup_{k \to \infty} f_k
\]

\[
\limsup_{k \to \infty} l_k \leq (I - \gamma P^\pi^*)^{-1} \limsup_{k \to \infty} f_k,
\]

since \( I - \gamma P^\pi^* \) is invertible. Finally, we only need to take the \( L_\infty \)-norm both sides and obtain,

\[
\limsup_{k \to \infty} \| l_k \| \leq \frac{\gamma}{1 - \gamma} \limsup_{k \to \infty} \| P^\pi_{k+1} (I - \gamma P^\pi_{k+1})^{-1} (I + \gamma P^\pi_k) + P^\pi^* \| \| e_k \|
\]

\[
\leq \frac{\gamma}{1 - \gamma} \left( \frac{1 + \gamma}{1 - \gamma} + 1 \right) \limsup_{k \to \infty} \| e_k \| = \frac{2\gamma}{(1 - \gamma)^2} \limsup_{k \to \infty} \| e_k \|.
\]
Outline

Performance Loss

Approximate Value Iteration

Approximate Policy Iteration
  Linear Temporal-Difference
  Least-Squares Temporal Difference
  Bellman Residual Minimization
Linear TD($\lambda$): the algorithm

Algorithm Definition

Given a linear space $\mathcal{F} = \{V_\alpha(x) = \sum_{i=1}^{d} \alpha_i \phi_i(x), \alpha \in \mathbb{R}^d\}$. Trace vector $z \in \mathbb{R}^d$ and parameter vector $\alpha \in \mathbb{R}^d$ initialized to zero. Generate a sequence of states $(x_0, x_1, x_2, \ldots)$ according to $\pi$. At each step $t$, the temporal difference is

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

and the parameters are updated as

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

where $\eta_t$ is learning step.
Linear TD(\(\lambda\)): approximation error

### Proposition (Tsitsiklis et Van Roy, 1996)

Let the learning rate \(\eta_t\) satisfy

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

We assume that \(\pi\) admits a *stationary distribution* \(\mu_\pi\) and that the features \((\phi_i)_{1 \leq k \leq K}\) are *linearly independent*. There exists a fixed \(\alpha^*\) such that

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

Furthermore we obtain

\[
\underbrace{\| V_{\alpha^*} - V^\pi \|_{2,\mu_\pi}}_{\text{approximation error}} \leq \frac{1 - \lambda \gamma}{1 - \gamma} \inf_{\alpha} \| V_\alpha - V^\pi \|_{2,\mu_\pi} \quad \underbrace{\text{smallest approximation error}}_{\text{smallest approximation error}}.
\]
Linear TD($\lambda$): approximation error

*Remark:* for $\lambda = 1$, we recover Monte-Carlo (or TD(1)) and the bound is the smallest!

*Problem:* the bound does not consider the variance (i.e., samples needed for $\alpha_t$ to converge to $\alpha^*$).
Linear TD(\(\lambda\)): implementation

- **Pros**: simple to implement, computational cost *linear* in \(d\).
- **Cons**: very sample *inefficient*, many samples are needed to converge.
Outline

Performance Loss

Approximate Value Iteration

Approximate Policy Iteration
  Linear Temporal-Difference
  Least-Squares Temporal Difference
  Bellman Residual Minimization
Least-squares TD: the algorithm

*Recall:* \( V^\pi = T^\pi V^\pi \).

*Intuition:* compute \( V = AT^\pi V \).

Focus on the \( L_{2,\mu} \)-weighted norm and projection \( \Pi_\mu \)

\[ \Pi_\mu g = \arg\min_{f \in \mathcal{F}} \| f - g \|_\mu. \]
Least-squares TD: the algorithm

By construction, the Bellman residual of $V_{TD}$ is orthogonal to $\mathcal{F}$, thus for any $1 \leq i \leq d$

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

and

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

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

$\Rightarrow \alpha_{TD}$ is the solution of a linear system of order $d$. 
Least-squares TD: the algorithm

Algorithm Definition
The LSTD solution $\alpha_{TD}$ can be computed by computing the matrix $A$ and vector $b$ defined as

$$A_{i,j} = \langle \phi_i, \phi_j - \gamma P^\pi \phi_j \rangle_\mu,$$

$$b_i = \langle \phi_i, r^\pi \rangle_\mu,$$

and then solving the system $A\alpha = b$. 
Least-squares TD: the approximation error

Problem: in general $\Pi_{\mu} T^\pi$ does not admit a fixed point (i.e., matrix $A$ is not invertible).

Solution: use the stationary distribution $\mu_{\pi}$ of policy $\pi$, that is

$$\mu_{\pi} P^\pi = \mu_{\pi}, \text{ and } \mu_{\pi}(y) = \sum_x p(y|x, \pi(x)) \mu_{\pi}(x)$$
Least-squares TD: the approximation error

Proposition

The Bellman operator $T^\pi$ is a \textit{contraction} in the weighted $L_{2,\mu_\pi}$-norm. Thus the joint operator $\Pi_{\mu_\pi} T^\pi$ is a contraction and it admits a unique \textit{fixed point} $V_{TD}$. Then

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

\textit{approximation error}  \hspace{1cm}  \textit{smallest approximation error}
Least-squares TD: the approximation error

Proof. We show that $\|P_\pi\|_{\mu_\pi} = 1$:

$$
\|P^\pi V\|_{\mu_\pi}^2 = \sum_x \mu_\pi(x) \left( \sum_y p(y|x, \pi(x)) V(y) \right)^2 \\
\leq \sum_x \sum_y \mu_\pi(x) p(y|x, \pi(x)) V(y)^2 \\
= \sum_y \mu_\pi(y) V(y)^2 = \|V\|_{\mu_\pi}^2.
$$

It follows that $T^\pi$ is a contraction in $L_{2,\mu_\pi}$, i.e.,

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

Thus $\Pi_{\mu_\pi} T^\pi$ is a composition of a non-expansion and a contraction in $L_{2,\mu_\pi}$, thus $V_{TD} = \Pi_{\mu_\pi} T^\pi V_{TD}$. 

Least-squares TD: the approximation error

**Proof.**
By Pythagorean theorem we have

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

but

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

Thus

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

which corresponds to eq. (?) after reordering.
Least-squares TD: the implementation

- Generate \((X_0, X_1, \ldots)\) from \textit{direct execution} of \(\pi\) and observes \(R_t = r(X_t, \pi(X_t))\)

- Compute estimates

\[
\hat{A}_{ij} = \frac{1}{n} \sum_{t=1}^{n} \phi_i(X_t)[\phi_j(X_t) - \gamma \phi_j(X_{t+1})],
\]

\[
\hat{b}_i = \frac{1}{n} \sum_{t=1}^{n} \phi_i(X_t)R_t.
\]

- Solve \(\hat{A}\alpha = \hat{b}\)

Remark:

- No need for a generative model.

- If the chain is ergodic, \(\hat{A} \to A\) et \(\hat{b} \to b\) when \(n \to \infty\).
Outline

Performance Loss

Approximate Value Iteration

Approximate Policy Iteration
- Linear Temporal-Difference
- Least-Squares Temporal Difference
- Bellman Residual Minimization
Bellman Residual Minimization (BRM): the idea

Let $\mu$ be a distribution over $X$, $V_{BR}$ is the minimum Bellman residual w.r.t. $T^\pi$

$$V_{BR} = \arg \min_{V \in \mathcal{F}} \| T^\pi V - V \|_{2,\mu}$$
Bellman Residual Minimization (BRM): the idea

The mapping $\alpha \rightarrow T^\pi V_\alpha - V_\alpha$ is affine. The function $\alpha \rightarrow \|T^\pi V_\alpha - V_\alpha\|^2_\mu$ is quadratic.

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

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

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

$$\begin{cases} A_{i,j} = \left\langle \phi_i - \gamma P^\pi \phi_i, \phi_j - \gamma P^\pi \phi_j \right\rangle_\mu, \\ b_i = \left\langle \phi_i - \gamma P^\pi \phi_i, r^\pi \right\rangle_\mu, \end{cases}$$
Bellman Residual Minimization (BRM): the idea

Remark: the system admits a solution whenever the features $\phi_i$ are linearly independent w.r.t. $\mu$

Remark: let $\{\psi_i = \phi_i - \gamma P^\pi \phi_i\}_{i=1\ldots d}$, 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 \( \mu_\pi \) is the *stationary policy* of \( \pi \), then \( \|P^\pi\|_{\mu_\pi} = 1 \) and

\[ \|(I - \gamma P^\pi)^{-1}\|_{\mu_\pi} = \frac{1}{1 - \gamma}, \text{ thus} \]

\[ \| V^\pi - V_{BR} \|_{\mu_\pi} \leq \frac{1 + \gamma}{1 - \gamma} \inf_{V \in \mathcal{F}} \| V^\pi - V \|_{\mu_\pi}. \]
BRM: the approximation error

**Proof.** We relate the Bellman residual to the approximation error as

\[ V^\pi - V = V^\pi - T^\pi V + T^\pi V - V = \gamma P^\pi (V^\pi - V) + T^\pi V - V \]

\[ (I - \gamma P^\pi) (V^\pi - V) = T^\pi V - V, \]

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 $\mu_\pi$, 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

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

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

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

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

⇒ minimizing \( \hat{B}(V) \) *does not* correspond to minimizing \( B(V) \) (even when \( n \to \infty \)).
BRM: the implementation

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

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

\( \Rightarrow \hat{B} \to B \text{ for } n \to \infty. \)
BRM: the implementation

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

$$
\hat{A}_{i,j} = \frac{1}{n} \sum_{t=1}^{n} \left[ \phi_i(X_t) - \gamma \phi_i(Y_t) \right] \left[ \phi_j(X_t) - \gamma \phi_j(Y'_t) \right],
$$

$$
\hat{b}_i = \frac{1}{n} \sum_{t=1}^{n} \left[ \phi_i(X_t) - \gamma \frac{\phi_i(Y_t) + \phi_i(Y'_t)}{2} \right] R_t.
$$
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 $\mu^\pi$. 
Bibliography I
Reinforcement Learning

Alessandro Lazaric
alessandro.lazaric@inria.fr
sequel.lille.inria.fr