Reinforcement Learning
Operations Research: Prediction and Planning

E. Le Pennec

ÉCOLE POLYTECHNIQUE

M2 DS - Fall 2022
RL: What Are We Going To See?

Outline

- Operations Research and MDP.
- Reinforcement learning and interactions.
- More tabular reinforcement learning.
- Reinforcement and approximation of value functions.
- Actor/Critic: a Policy Point of View
How to find the best policy knowing the MDP?

- Is there an optimal policy?
- How to estimate it numerically?

- Finite states/actions space assumption (tabular setting).
- Focus on iterative methods using value functions (dynamic programming).
- Policy deduced by a statewise optimization of the value function over the actions.
- Most results for the discounted setting.
**Known MDP model**

**Focus on the finite horizon setting**

\[ G^T_t = \sum_{t'=t+1}^{T} R_{t'} \]

and the discounted setting:

\[ G^\gamma_t = \sum_{t'=t+1}^{\infty} \gamma^{t'-(t+1)} R_{t'} \]

- We will later consider the other settings.
Policy

- Finite horizon: emphasis on Markovian policies
  \[ \Pi_t(A_t = a_t) = \pi_t(A_t = a_t | S_t = s_t) = \pi_t(a_t | s_t) \]

- Discounted return: emphasis on stationary Markovian policies
  \[ \Pi_t(A_t = a_t) = \pi(A_t = a_t | S_t = s_t) = \pi(a_t | s_t) \]
How to efficiently evaluate the quality of a policy

\[ v_{t,\pi}(s) = \mathbb{E}_\pi \left[ \sum_{t'=t+1}^{T} \gamma^{t'-t-1} R_{t'} \big| S_t = s \right] \]

when we can ensure that the sum is finite?

- \( v_{t,\pi} \) independent of \( t \) in the discounted setting if the policy is stationary.
Policy

- How to find a policy $\pi$ such that
  \[ \sum_{s,t} \mu(s, t) v_{t, \Pi}(s) \]
  is as large as possible?
- Emphasis on $\mu(s, t) = 0$ if $t \neq 0$ and $\mu(s, 0) = \mathbb{P}_0(S_0 = s_0)$. 

Planning

MDP

Model MDP

Model Env.

Model Agent

Agent Policy

Value Functions

Policy

Agent Policy

Final Policy

Behavior Policy

Real Time Policy

How to find a policy $\pi$ such that

[Equation]

is as large as possible?

Emphasis on $\mu(s, t) = 0$ if $t \neq 0$ and $\mu(s, 0) = \mathbb{P}_0(S_0 = s_0)$. 

Outline

1. Prediction and Bellman Equation
2. Prediction by Dynamic Programming and Contraction
3. Planning, Optimal Policies and Bellman Equation
4. Linear Programming
5. Planning by Value Iteration
6. Planning by Policy Iteration
7. Optimization Interpretation
8. Approximation and Stability
9. Generalized Policy Iteration
10. Infinite, Episodic and Average setting
11. References
Bellman Equation

\[ \nu_{t, \pi}(s) = \sum_a \pi_t(a|s) \sum_{s', r} p(s', r|s, a) \left( r + \gamma \nu_{t+1, \pi}(s') \right) \]

\[ = \sum_a \pi_t(a|s) r(s, a) + \gamma \sum_{s'} \sum_a p(s'|s, a) \pi_t(a|s) \nu_{t+1, \pi}(s') \]

Bellman Equation

- Link between \( \nu_{t, \pi} \) and \( \nu_{t+1, \pi} \).
- Straightforward consequence of

\[ G_t = \sum_{t'=t+1}^T \gamma^{t'-(t+1)} R_{t'} = R_{t+1} + \gamma \sum_{t'=t+2}^T \gamma^{t'-(t+2)} R_{t'} = R_{t+1} + \gamma G_{t+1} \]

and thus

\[ \mathbb{E}[G_t|S_t = s] = \mathbb{E}[R_{t+1}|S_t = s] + \gamma \mathbb{E}[\mathbb{E}[G_{t+1}|S_{t+1}]|S_t = s] \]
Bellman Operator

\( \mathcal{T}^{\pi_t} : \mathbb{R}^{|S|} \rightarrow \mathbb{R}^{|S|} \)

\[
\mathcal{T}^{\pi_t} v(s) = \sum_a \pi_t(a|s) r(s, a) + \gamma \sum_{s'} \sum_a \pi_t(a|s) v(s') 
\]

- Affine operator from the space of state value functions to the space of state value functions.
- By construction,

\[
v_t, \Pi = \mathcal{T}^{\pi_t} v_{t+1}, \Pi
\]

- \( r_{\pi_t} \) is the vector of average immediate rewards using policy \( \pi_t \) while \( P^{\pi_t} \) is the one step state transition matrix using policy \( \pi_t \).
Outline

1 Prediction and Bellman Equation
2 Prediction by Dynamic Programming and Contraction
3 Planning, Optimal Policies and Bellman Equation
4 Linear Programming
5 Planning by Value Iteration
6 Planning by Policy Iteration
7 Optimization Interpretation
8 Approximation and Stability
9 Generalized Policy Iteration
10 Infinite, Episodic and Average setting
11 References
Finite Horizon: Naive Approach

\[
v_{t,\Pi}^T(s) = \sum_{a_t, r_{t+1}, s_{t+1}, \ldots, r_T} \left( \sum_{t' = t+1}^T r_{t'} \right) \prod_{i=0}^{T-t-1} \mathbb{P}_\Pi(A_t = a_t \ldots, R_T = r_T | S_t = s)
\]

\[
= \sum_{a_t, r_{t+1}, s_{t+1}, \ldots, r_T} \left( \sum_{t' = t+1}^T r_{t'} \right) \pi_t(a_t | s) \times \cdots \times p(s_T, r_T | s_{T-1}, a_{T-1})
\]

Finite Horizon: Naive Approach

- Exhaustive exploration of the trajectories.
- Complexity of order \((|A| \times |S| \times |R|)^{T-t}\) for the value function at time \(t\).
- Complexity can be reduced to \((|A| \times |S|)^{T-t}\) by noticing that

\[
v_{t,\Pi}^T(s) = \sum_{a_t, s_{t+1}, \ldots, s_{T-1}, a_{T-1}} \left( \sum_{t' = t+1}^T r(s_t, a_t) \right) \pi_t(a_t | s) \times \cdots \times p(s_T | s_{T-1}, a_{T-1})
\]
Finite Horizon: Recursive Prediction

\[ v_{T, \pi}^T = 0 \]
\[ v_{t-1, \pi}^T = \mathcal{T}^{\pi_{t-1}} v_{t, \pi}^T \]

After time \( T \), the finite horizon return \( G_t^T = 0 \) hence \( v_{T, \pi}^T = 0 \) whatever the policy.

The Bellman equation yields second equation.

Equivalent rewriting

\[ v_{t-1, \pi}(s) = r_{\pi_{t-1}}(s) + \sum_{s'} P_{\pi_{t-1}}(s, s') v_{t}^T \]

Complexity of order only \( T \times |S|^2(|A| + |S|) \) to compute all the value functions.
Finite Horizon: Value Iteration

input: MDP model \((S, A, R, P)\) and policy \(\Pi\)
parameter: Horizon \(T\)
init: \(v_T^T(s) = 0 \forall s \in S, t = T\)
repeat
\[
t \leftarrow t - 1
\]
for \(\forall s \in S\) do
\[
v_t^T(s) \leftarrow \sum_{a \in A} \pi_t(a|s) \left( r(s, a) + \gamma \sum_{s' \in S} p(s'|s, a) v_{t+1}^T(s') \right)
\]
end
until \(t = 0\)
output: Value functions \(v_t^T\)

Most classical formulation
**Discounted: Naive Approach**

\[
 v_{t,\pi}^\gamma(s) = \sum_{t'=t+1}^{\infty} \gamma^{t'-(t+1)} \mathbb{E}_\pi[R_{t'}|S_t = s] \simeq \sum_{t'=t+1}^{T} \gamma^{t} \mathbb{E}_\pi[R_{t'}|S_t = s] = v_{t,\pi}^{\gamma,T}(s)
\]

\[
 v_{t,\pi}^{\gamma,T}(s) = \sum_{a_t,s_{t+1},\ldots,s_{t-1},a_{t-1}} \left( \sum_{t'=t+1}^{T} \gamma^{t'-(t+1)} r(s_t, a_t) \right) \pi_t(a_t|s) \times \cdots \times p(s_T|s_{t-1}, a_{t-1})
\]

**Naive approach**

- Exhaustive exploration of truncated trajectories.
- Back to the finite horizon setting...

**Prop:** Control on the error as
\[
 |v_{\pi}^\gamma - v_{t,\pi}^{\gamma,T}|_\infty \leq \frac{\gamma^{T+1-t}}{1-\gamma} \max_{r \in \mathcal{R}} |r|
\]

- Relation between the error \( \epsilon \simeq \gamma^{T-t} \) and the numerical complexity
  \( C = (|\mathcal{A}| \times |S|)^{T-t} \) of order \( C \simeq \epsilon^{-1} \).
Discounted: Recursive Prediction with Naive Initialization

\[ v_{T,\pi}^\gamma \simeq v_{T',\pi}^\gamma = \tilde{v}_{T,\pi} \]

\[ v_{t-1,\pi}^\gamma = \tau^{\pi_{t-1}} v_{t,\pi}^\gamma \simeq \tilde{v}_{t-1,\pi} = \tau^{\pi_{t-1}} \tilde{v}_{t,\pi} \]

**Recursive Prediction**

- Requires an initialization at time \( T \) with a horizon \( T' \).
- The Bellman equation yields the second equation.
- Complexity of order only \( T \times |S|^2 (|A| + |S|) \) to compute all the value functions after the initialization of cost \( (|A| \times |S|)^{T'-T} \).
- **Prop:** If the approximation error between \( v_{T,\pi}^\gamma \) and \( v_{T',\pi}^\gamma \) is bounded by \( \epsilon \) then
  \[ \| v_{t,\pi}^\gamma - \tilde{v}_{t,\pi} \|_\infty \leq \gamma^{T-t} \epsilon, \quad \forall t \leq T \]
Prediction by Dynamic Programming and Contraction

Discounted and stationary: Bellman Equation

$$\Pi = (\pi, \pi, \ldots, \pi)$$

$$v_\Pi = T^\pi v_\Pi$$

$$v_\Pi(s) = \sum_a \pi(a|s) r(s, a) + \gamma \sum_{s'} \sum_a p(s'|s, a) \pi(a|s) v_\Pi(s')$$

**Bellman Equation**

- Time independent value function $v_\Pi$.

- **Prop:** Unique solution of the linear equation $v_\Pi = T^\pi v_\Pi$

- Complexity of order $(|A| + |S|) \times |S|^2$ to obtain the solution.
Discounted and stationary: Recursive Implementation

\[ v_\Pi = T^\pi v_\Pi \]

\[ v_{k+1} = T^\pi v_k \quad \text{with arbitrary } v_0 \]

**Bellman Iteration**

- **Prop:** Unique fixed point of the Bellman operator \( v \mapsto T^\pi v \).
- **Prop:** The iterates \( v_{k+1} = T^\pi v_k \) converges toward \( v_\Pi \) and
  \[ \| v_k - v_\Pi \|_\infty \leq \gamma^k \| v_0 - v_\Pi \|_\infty \]
- Complexity of order \((k + |A||S|^2)\) to obtain the \( k \)th iterate.
- Exponential decay of the error with respect to the complexity.
Bellman Operator and Contraction

\[ \| T^\pi v - T^\pi v' \|_\infty \leq \gamma \| v - v' \|_\infty \]

**Proof**

- By definition
  \[ \| T^\pi v - T^\pi v' \|_\infty = \gamma \| P^\pi (v - v') \|_\infty \]
- It suffices then to notice that \( P^\pi \) is a transition matrix, so that
  \[ \sum_j P^\pi_{i,j} = 1 \]
  and thus \[ \left| \sum_j P^\pi_{i,j} z_j \right| \leq \max |z_j| \]

**Consequences**

- Unicity of the solution of \( T^\pi v = v \).
- Linear decay \( \gamma^k \) of the error with the iterates.
\[ Z_{\pi} = n_{\pi} + \chi P_{\pi} \]

\[ Z'_{\pi} = n'_{\pi} + \chi P_{\pi}' \]

\[ Z_{\pi} - Z_{\pi}' = (n_{\pi} + \chi P_{\pi}) - (n_{\pi} + \chi P_{\pi}) \]

\[ = \chi P_{\pi} (n_{\pi} - n_{\pi}') \]

\[ = \chi P_{\pi} ||n_{\pi} - n_{\pi}'||_\infty \]

\[ \leq \sum_i p_{\pi}(s_i) \rho(s_{i+1}) \pi(s_{i+1}) \pi(s_i) \sum_{s'} \pi'(ds') \rho(s'_{i+1}) \pi(s'_{i+1}) \]

\[ \leq \sum_i p_{\pi}(s_i) \rho(s_{i+1}) (\pi(s_i) - \pi(s_i)) \]

\[ \leq \sum_i p_{\pi}(s_i) ||\pi(s_i) - \pi(s_i)||_\infty \]

\[ \leq \sum_i p_{\pi}(s_i) ||v(s_i) - v(s_i)||_\infty \]

\[ \leq \sum_i p_{\pi}(s_i) \rho(s_{i+1}) ||v(s_{i+1}) - v(s_{i+1})||_\infty \]
Bellman Operator and Bellman Equation Solution

\[ v_\pi = \left( \sum_{k=0}^{\infty} \gamma^k \left( P^\pi \right)^k \right) r_\pi \]

A Closed Formula for the State Value Function

- \( v_\pi = T^\pi v_\pi \Leftrightarrow (I - \gamma P^\pi) v_\pi = r_\pi \)
- As \( P^\pi \) is a transition matrix, its eigenvalues are smaller than 1 and thus \((I - \gamma P^\pi)\) is invertible of inverse

\[ (I - \gamma P^\pi)^{-1} = \sum_{k=0}^{\infty} \gamma^k \left( P^\pi \right)^k \]

- Could have been obtained without the Bellman equation as the \( \left( (P^\pi)_s^{s'} \right) \) is, by construction, the probability of being at state \( s' \) at time \( k \) starting from \( s \) at time 0 and following \( \Pi \).
Discounted and stationary: Value Iteration

**Discounted: Prediction by Value Iteration**

**input:** MDP model $\langle (S, A, R), P \rangle$, discount factor $\gamma$, and stationary policy $\pi$

**init:** $\tilde{v}(s) \forall s \in S$

**repeat**

- $\tilde{v}_{\text{prev}} \leftarrow \tilde{v}$
- **for** $s \in S$ **do**
  - $\tilde{v}(s) \leftarrow \sum_{a \in A} \pi(a|s) \left( r(s, a) + \gamma \sum_{s' \in S} p(s'|s, a)\tilde{v}_{\text{prev}}(s') \right)$

**output:** Value function $\tilde{v}$

- When to stop?
Discounted and stationary: Value Iteration

**Discounted: Prediction by Value Iteration**

**input:** MDP model \( \langle S, A, R, P \rangle \), discount factor \( \gamma \), and stationary policy \( \pi \)

**parameter:** \( \delta > 0 \) as accuracy termination threshold

**init:** \( \tilde{v}(s) \) \( \forall s \in S \)

**repeat**

\[
\tilde{v}_{\text{prev}} \leftarrow \tilde{v} \\
\Delta \leftarrow 0 \\
\text{for } s \in S \text{ do} \\
\quad \tilde{v}(s) \leftarrow \sum_{a \in A} \pi(a|s) \left( r(s, a) + \gamma \sum_{s' \in S} p(s'|s, a) \tilde{v}_{\text{prev}}(s') \right) \\
\quad \Delta \leftarrow \max (\Delta, |\tilde{v}(s) - \tilde{v}_{\text{prev}}(s)|) \\
\text{end}
\]

**until** \( \Delta < \delta \)

**output:** Value function \( \tilde{v} \)

- **Prop:** when the algorithms stops

\[
\|\tilde{v} - v_\Pi\|_\infty \leq \frac{2\delta}{1-\gamma}
\]
Discounted and stationary: Value Iteration

---

**Discounted: Prediction by Value Iteration - Gauss-Seidel Version**

**Input:** MDP model \( \langle (S, A, R), P \rangle \), discount factor \( \gamma \), and stationary policy \( \pi \)

**Parameter:** \( \delta > 0 \) as accuracy termination threshold

**Init:** \( \tilde{v}(s) \forall s \in S \)

**Repeat**
- \( \Delta \leftarrow 0 \)
- for \( s \in S \) do
  - \( \tilde{v}_{\text{prev}} \leftarrow \tilde{v}(s) \)
  - \( \tilde{v}(s) \leftarrow \sum_{a \in A} \pi(a|s) \left( r(s, a) + \gamma \sum_{s' \in S} p(s'|s, a)\tilde{v}(s') \right) \)
  - \( \Delta \leftarrow \max(\Delta, |\tilde{v}(s) - \tilde{v}_{\text{prev}}|) \)
- until \( \Delta < \delta \)

**Output:** Value function \( \tilde{v} \)

- Gauss-Seidel variation mostly used in practice.
- No need to store the previous value function.
Outline

1. Prediction and Bellman Equation
2. Prediction by Dynamic Programming and Contraction
3. Planning, Optimal Policies and Bellman Equation
4. Linear Programming
5. Planning by Value Iteration
6. Planning by Policy Iteration
7. Optimization Interpretation
8. Approximation and Stability
9. Generalized Policy Iteration
10. Infinite, Episodic and Average setting
11. References
### Optimal Policy

- An optimal policy $\Pi^*$ should be better than any other policies:

$$\forall s, \forall t, \nu_{t,\Pi^*}(s) = \sup \nu_{t,\Pi}(s)$$

### Several Questions

- Do this policy exists?
- Is it unique?
- How to characterize it?
- How to obtain it?

- Even the $\sup$ above could be an issue if it is not attained!
Finite Horizon and Optimal Policy

Explicit Recursive Solution

- After horizon $T$, any policy leads to a 0 return.
- At time $T-1$,
  - the total return $G_T$ is the immediate return at time $T$ and thus
    \[ v_{T, \pi^*}(s) = \sup_{\pi(a|s)} \sum_a \pi(a|s) r(a, s) = \sup_a r(a, s) \]
  - the optimal policy $\pi^*_T$ exists and is deterministic.
- By recursion,
  - the total return at time $t-1$ is the immediate return at time $t$ plus the total return at time $t-1$ and thus
    \[ v_{t-1, \pi^*}(s) = \sup_{\pi(a|s)} \sum_a \pi(a|s) \left( r(a, s) + \sum_{s'} p(s'|s, a)v_{t, \pi^*} \right) \]
    \[ = \sup_a \left( r(a, s) + \sum_{s'} p(s'|s, a)v_{t, \pi^*} \right) \]
  - the optimal policy $\pi^*_{t-1}$ exists and is deterministic.
Discounted Setting and Optimal Stationary Policy

Heuristic

- Optimal policy: $v_\Pi^*(s) = \sup_\pi v_\Pi(s)$
- Stationary solution:
  $$v_\Pi^*(s) = \sup_\pi (T^\pi v_\Pi^*) (s)$$
  $$= \sup_{\pi_t(\cdots|s)} \sum_a \pi(a|s) \left( r(a, s) + \sum_{s'} p(s'|s, a) v_\Pi^*(s') \right)$$
  $$= \sup_a \left( r(a, s) + \sum_{s'} p(s'|s, a) v_\Pi^*(s') \right)$$
- Optimal deterministic policy: $\pi^*(s) \in \arg\max (r(a, s) + \sum_{s'} p(s'|s, a) v_\Pi^*(s'))$.

- Is everything well defined? Yes but one has to be more cautious!
## Optimal Value Function and Bellman Operator

### Optimal Value Function
- Optimal value function: \( v_*(s) = \sup_\Pi v_\Pi(s) \)
- Defined state by state so that it is not necessarily attained by a single \( \Pi^* \)

### Optimal Bellman operator
- Similar to the Bellman operator but do not depend on a policy:
  \[
  T^*v(s) = \sup_a \left( r(a, s) + \sum_{s'} p(s'|s, a) v(s') \right)
  \]

### Link between the two
- \( v \geq T^*v \) implies \( v \geq v_* \).
- \( v \leq T^*v \) implies \( v \leq v_* \).
$n \geq 2^n \Rightarrow n \geq n^x$

$n^z(s) = \sup \ n^z_T(s)$

$\Pi = (T_0, \ldots, T_0) \Rightarrow \Pi^\pi \sim \Sigma_T \leq n$

$\Rightarrow n^x_T = 2^{n^x_T} \Rightarrow n^x_T = 2^0 (2^0 \ldots (2^0 n^x_T))$

$n \Rightarrow n \geq 2^n \Rightarrow 2^n n^x$

$n \geq 2^0 (2^0 \ldots (2^0 n^x_T))$

$n = 2^0 (2^0 \ldots (2^0 n^x_T) + |2^0 \ldots (2^0 n^x_T) (2^0 \ldots (2^0 n^x_T)| (2^n n^x_T))$
Optimal Value Function and Bellman Operator

Bellman Operator and Fixed Point

- **Prop:** $T^*$ is a $\gamma$-contraction for the sup-norm and thus it exists a unique $v_{**}$ such that $v_{**} = T^*v_{**}$.

Fixed Point and Optimal Value Function

- **Prop:** $v_* = v_{**}$ and is thus the unique fixed point of $T^*$.
- **Proof:** $v_{**} = T^*v_{**}$ and thus $v_{**} = v_*$ according the link between the optimal value function and the Bellman operator.

Does this mean something about policies?
**Bellman Operator and Policy**

- **Prop:** For any \( v \), any policy \( \pi_v \) satisfying

\[
\pi_v(s) \in \arg \max_a \left( r(a, s) + \sum_{s'} p(s'|s, a) v(s') \right)
\]

is such that \( T^* v(s) = \sup_\pi T^{\pi} v(s) = T^{\pi_v} v(s) \)

**Bellman Operator and Optimal Policy**

- **Prop:** Any stationary policy \( \pi_\star \) satisfying

\[
\pi_\star(s) \in \arg \max_a \left( r(a, s) + \sum_{s'} p(s'|s, a) v_\star(s') \right)
\]

is optimal.

- **Proof:** Indeed by construction, \( T^* v_\star = T^{\pi_\star} v_\star \) and thus, as \( T^* v_\star = v_\star \), \( v_{\pi_\star} = v_\star \).
Optimal Policy and Bellman Operator

Summary

- It exists a unique $v_\star$ such that $\mathcal{T}^*v_\star = v_\star$
- $\forall s, v_\star(s) = \sup_\pi v_\pi(s)$
- Any policy $\pi_\star$ satisfying:

$$\forall s, \pi_\star(s) \in \arg\max_a \left( r(a, s) + \sum_{s'} p(s'|s, a) v_\star(s') \right)$$

is optimal as $\forall s, v_{\pi_\star}(s) = v_\star(s) = \sup_\pi v_\pi(s)$

- Existence result but not (yet) a constructive algorithm!
Outline

1. Prediction and Bellman Equation
2. Prediction by Dynamic Programming and Contraction
3. Planning, Optimal Policies and Bellman Equation
4. Linear Programming
5. Planning by Value Iteration
6. Planning by Policy Iteration
7. Optimization Interpretation
8. Approximation and Stability
9. Generalized Policy Iteration
10. Infinite, Episodic and Average setting
11. References
\[ v_\pi = \mathcal{T}^\pi v_\pi \quad v_* = \mathcal{T}^* v_* \]

**Explicit Resolution of the Equations?**

- **Prediction:**
  - Simple linear system for \( v_\pi \).
  - Already mentioned before.
  - Complexity of order \((|A| + |S|)|S|^2\).

- **Planning:**
  - More complex linear programming system for \( v_* \) due to the max operator.
  - Optimal policy easily deduced from \( v_* \).
  - Complexity of order \((|A||S|)^3\).
Linear Programming

From $\forall s$, $v(s) = \sup_a r(s, a) + \gamma \sum_{s'} p(s'|s, a) v(s')$

to $\min v \sum_s \mu(s) v(s)$

such that $\forall (s, a), v(s) \geq r(s, a) + \gamma \sum_{s'} p(s'|s, a) v(s')$

Different formulations but same solution

- Using $v \geq T^* v \iff v \geq v_*$, the condition implies $v \geq v_*$

- Now for any $\mu$ satisfying $\mu(s) > 0$, $\sum_s \mu(s) v(s) \geq \sum_s \mu(s) v_*(s)$ as soon as the condition is satisfied, hence $v_*$ is a solution.

- If for any state $v(s) > v_*(s)$ then $\sum_s \mu(s) v(s) > \sum_s \mu(s) v_*(s)$ and thus $v_*$ is the unique minimizer.
Primal Problem

Primal: \( \min_v \sum_s \mu(s)v(s) \)

such that \( \forall (s, a), v(s) \geq r(s, a) + \gamma \sum_{s'} p(s'|s, a)v(s') \)

Some properties

- Can be solved with a linear programming solver.
- Unicity of solution (and thus independence with respect to \( \mu \)) can be proved without using \( v_* \).

**Proof:** let \( v_1 \) a solution for \( \mu_1 \) and \( v_2 \) a solution for \( \mu_2 \) then \( \min(v_1, v_2) \) satifies the constraints. Furthermore if exists \( v_2(s) < v_1(s) \) then \( \min(v_1, v_2) \) is a strictly better solution for \( \mu_2 \) which is impossible.
Dual Problem

Primal: \( \min \sum_{s} \mu(s) v(s) \)

such that \( \forall (s, a), v(s) \geq r(s, a) + \gamma \sum_{s'} p(s'|s, a) v(s') \)

Dual: \( \max_{\lambda(s,a) \geq 0} \sum_{s,a} \lambda(s, a) r(s, a) \)

such that \( \forall s, \sum_{a} \lambda(s, a) = \mu(s) + \gamma \sum_{s',a} p(s|s', a) \lambda(s', a) \)

Derivation

- Usual derivation through the Lagrangian:

\[
L(v, \lambda) = \sum_{s} \mu(s) v(s) + \sum_{s,a} \lambda(s, a) \left( r(s, a) + \gamma \sum_{s',a} p(s|s', a) v(s') - v(s) \right)
\]

- Strong duality as Slater condition holds when \( \gamma < 1 \) with \( v = \frac{1+\gamma}{1-\gamma} \max_{s,a} r(s, a) \).
## Dual and Interpretation

**Dual:**
\[
\max_{\lambda(s,a) \geq 0} \sum_{s,a} \lambda(s, a) r(s, a)
\]

such that \(\forall s, \sum_a \lambda(s, a) = \mu(s) + \gamma \sum_{s', a} p(s'|s', a) \lambda(s', a)\)

**Interpretation:**
\[
\max_{\pi} \sum_{k=0}^{\infty} \gamma^k \sum_{s,a} \mathbb{P}(S_t = s, A_t = a | S_0 \sim \mu, \pi) r(s, a)
\]

### Interpretation in terms of policy

- For any feasible \(\lambda\), define \(u(s) = \sum_a \lambda(s, a)\) and the policy \(\pi(a|s) = \lambda(s, a)/u(s)\).
- **Prop:**\( u = (\text{Id} - \gamma P^\pi) \mu = \sum_{k=0}^{\infty} \gamma^k (P^\pi)^k \mu \).
- **Prop:** \(\lambda(s, a) = \pi(a|s)u(s) = \sum_{k=0}^{\infty} \gamma^k \mathbb{P}(S_t = a, A_t = a | S_0 \sim \mu, \pi)\)
- Conversely for any \(\pi\) they is a feasible \(\lambda\).
- Any optimal \(\lambda^*_\star\) (and thus policy) satisfies \(\lambda^*_\star(s,a) = 0\) if \(v^*_\star(s) > r(s, a) + \gamma \sum_{s'} p(s'|s,a)v^*_\star(s')\) (optimal policy support).
Outline

1. Prediction and Bellman Equation
2. Prediction by Dynamic Programming and Contraction
3. Planning, Optimal Policies and Bellman Equation
4. Linear Programming
5. Planning by Value Iteration
6. Planning by Policy Iteration
7. Optimization Interpretation
8. Approximation and Stability
9. Generalized Policy Iteration
10. Infinite, Episodic and Average setting
11. References
Finite Horizon: Planning by Value Iteration

input: MDP model $\langle (S, A, R), P \rangle$

parameter: Horizon $T$

init: $v_T^T(s) = 0 \forall s \in S$, $t = T$

repeat

\[
\begin{align*}
    t &\leftarrow t - 1 \\
    \text{for } s \in S \text{ do} \\
    &v_t^T(s) \leftarrow \max_{a \in A} \left( r(s, a) + \gamma \sum_{s' \in S} p(s'|s, a) v_{t+1}^T(s') \right) \\
    \text{end} \\
\end{align*}
\]

until $t = 0$

output: Deterministic policy $\pi_t(s) \in \arg\max_{a \in A} \left( r(s, a) + \gamma \sum_{s' \in S} p(s'|s, a) v_{t+1}^T(s') \right)$

- Algorithm used to prove the existence of an optimal policy.
- No necessarily unique as $\arg\max$ may not be unique.
Optimal Value Function, Fixed Point and Contraction

\( v_\star = T^* v_\star \) and \( \| T^* v - T^* v' \|_\infty \leq \gamma \| v - v' \|_\infty \)

\[ \implies v_{k+1} = T^* v_k \rightarrow v_\star \]

**Bellman Operator**

- Properties of Optimal Bellman Operator:
  - \( v_\star \) is a fixed point of \( T^* \).
  - \( T^* \) is a \( \gamma \)-contraction for the \( \| \cdot \|_\infty \) norm.
- Classical fixed point theorem setting.
- Practical algorithm to approximate \( v_\star \).
### Discounted: Value Iteration Planning

**input:** MDP model $\langle (S, A, R), P \rangle$, and discount factor $\gamma$

**parameter:** $\delta > 0$ as accuracy termination threshold

**init:** $\tilde{v}(s) \forall s \in S$

**repeat**

- $\tilde{v}_{prev} \leftarrow \tilde{v}$
- $\Delta \leftarrow 0$

  **for** $s \in S$ **do**
  
  - $\tilde{v}(s) \leftarrow \max_{a \in A} r(s, a) + \gamma \sum_{s' \in S} p(s' | s, a) \tilde{v}_{prev}(s')$
  
  - $\Delta \leftarrow \max (\Delta, |\tilde{v}(s) - \tilde{v}_{prev}(s)|)$

**until** $\Delta < \delta$

**output:** Value function $\tilde{v}$

- Same convergence criterion (and similar proof) than in the planning case.
- Which policy?
Discounted: Value Iteration Planning

**input:** MDP model $\langle (S, A, \mathcal{R}), P \rangle$, and discount factor $\gamma$

**parameter:** $\delta > 0$ as accuracy termination threshold

**init:** $\tilde{v}(s) \forall s \in S$

**repeat**

- $\tilde{v}_{\text{prev}} \leftarrow \tilde{v}$
- $\Delta \leftarrow 0$

  **for** $s \in S$ **do**

  - $\tilde{v}(s) \leftarrow \max_{a \in A} r(s, a) + \gamma \sum_{s' \in S} p(s' | s, a) \tilde{v}_{\text{prev}}(s')$
  - $\Delta \leftarrow \max (\Delta, |\tilde{v}(s) - \tilde{v}_{\text{prev}}(s)|)$

**end**

**until** $\Delta < \delta$

**output:** Deterministic policy $\tilde{\pi}(s) \in \argmax_a r(s, a) + \gamma \sum_{s' \in S} p(s' | s, a) \tilde{v}(s')$

- Natural idea: define a policy using the argmax of the existence proof.
- Do we have a convergence guarantee on the resulting policy?
Value and argmax Policy

\[ \tilde{\pi}(s) \in \arg\max_a r(s, a) + \gamma \sum_{s'} p(s'|s, a) \tilde{v}(s') \]

\[ \implies \| v_{\tilde{\pi}} - v_* \|_\infty \leq \frac{2\gamma}{1 - \gamma} \| \tilde{v} - v_* \|_\infty \]

Value and argmax Policy

- Bound on the loss of the final policy!
- Rely on the fact that, by construction, \( \mathcal{T} \tilde{\pi} \tilde{v} = \mathcal{T}^* \tilde{v} \)
- **Proof:**

\[ \| v_{\tilde{\pi}} - v_* \|_\infty = \| \mathcal{T} \tilde{\pi} v_{\tilde{\pi}} - \mathcal{T} \tilde{\pi} \tilde{v} + \mathcal{T}^* \tilde{v} - \mathcal{T}^* v_* \|_\infty \]
\[ \leq \| \mathcal{T} \tilde{\pi} v_{\tilde{\pi}} - \mathcal{T} \tilde{\pi} \tilde{v} \|_\infty + \| \mathcal{T}^* \tilde{v} - \mathcal{T}^* v_* \|_\infty \]
\[ \leq \gamma \| v_{\tilde{\pi}} - \tilde{v} \|_\infty + \gamma \| \tilde{v} - v_* \|_\infty \]
\[ \leq \gamma \| v_{\tilde{\pi}} - v_* \|_\infty + 2\gamma \| \tilde{v} - v_* \|_\infty \]
Valued Iteration Algorithm

Discounted: Value Iteration Planning

input: MDP model $\langle (S, A, R), P \rangle$, and discount factor $\gamma$

discount factor $\gamma$

parameter: $\delta > 0$ as accuracy termination threshold

init: $\tilde{v}(s) \forall s \in S$

repeat

$\tilde{v}_{\text{prev}} \leftarrow \tilde{v}$

$\Delta \leftarrow 0$

for $s \in S$ do

$\tilde{v}(s) \leftarrow \max_{a \in A} r(s, a) + \gamma \sum_{s' \in S} p(s' | s, a) \tilde{v}_{\text{prev}}(s')$

$\Delta \leftarrow \max (\Delta, |\tilde{v}(s) - \tilde{v}_{\text{prev}}(s)|)$

end

until $\Delta < \delta$

output: Deterministic policy $\tilde{\pi}(s) \in \arg\max_{a} r(s, a) + \gamma \sum_{s' \in S} p(s' | s, a) \tilde{v}(s')$

Prop: $\|v_{\tilde{\pi}} - v_\star\|_\infty \leq \frac{4\gamma \delta}{1 - \gamma}$
From State Value to State-Action Value Functions

\( v_\pi(s) = \mathbb{E}_\pi \left[ \sum_k \gamma^k R_t | S_0 = s \right] \)

\( T^\pi v(s) = \sum_a \pi(a | s) \left( r(s, a) + \sum_{s'} p(s' | s, a) v(s') \right) \)

\( T^\pi v(s) = \sum_a \pi(a | s) \left( r(s, a) + \sum_{s'} p(s' | s, a) \sum_{a'} \pi(a' | s') q(s', a) \right) \)

\( T^* v(s) = \max_a r(s, a) + \sum_{s'} p(s' | s, a) v(s') \)

\( T^* v(s) = \max_a r(s, a) + \sum_{s'} p(s' | s, a) \max_{a'} q(s', a) \)

\( q_\pi(s, a) = \mathbb{E}_\pi \left[ \sum_k \gamma^k R_t | S_0 = s, A_0 = a \right] \)

\( T^\pi q(s, a) = r(s, a) + \sum_{s'} p(s' | s, a) \sum_{a'} \pi(a' | s') q(s', a) \)

\( T^* q(s, a) = r(s, a) + \sum_{s'} p(s' | s, a) \max_{a'} q(s', a) \)

Two equivalent point of view?

- Everything could have been defined using the state-action point of view.
- Knowing \( v_\pi \) is equivalent to knowing \( q_\pi \) as

\( v_\pi(s) = \sum_a \pi(s | a) q_\pi(s, a) \) and \( q_\pi(s, a) = r(s, a) + \sum_{s'} p(s' | s, a) v_\pi(s') \).
State-Action Bellman Operators

\[ T^\pi q(s, a) = r(s, a) + \sum_{s'} p(s'|s, a) \sum_a \pi(a|s') q(s', a) \]

\[ T^* q(s, a) = r(s, a) + \sum_{s'} p(s'|s, a) \max_a q(s', a) \]

Properties

- **Prop:** \( T^\pi \) and \( T^* \) are \( \gamma \) contractions for the \( \| \cdot \|_\infty \) norm.
- **Prop:** \( q_\pi \) is the unique solution of \( T^\pi q = q \)
- **Prop:** \( q_* \) defined \( q_*(s, a) = \sup_\pi q_\pi(s, a) \) is the unique solution of \( q = T^* q \) and is attained for any policy \( \pi_* \) satisfying \( \pi_*(s) \in \arg\max q_*(s, a) \).
- **Prop:** Any such policy satisfies: \( v_{\pi_*}(s) = q_{\pi_*}(s, \pi_*(s)) = v_*(s) \).
State-Action Value Iteration Algorithm

Discounted: Planning by State-Action Value Iteration

input: MDP model $\langle (S, A, R), P \rangle$, and discount factor $\gamma$

parameter: $\delta > 0$ as accuracy termination threshold

init: $\tilde{q}(s, a) \forall (s, a) \in S \times A$

repeat
\[
\tilde{q}_{\text{prev}} \leftarrow \tilde{q} \\
\Delta \leftarrow 0 \\
\text{for } s \in S \text{ do} \\
\quad \text{for } a \in A \text{ do} \\
\quad \quad \tilde{q}(s, a) \leftarrow \left( r(s, a) + \gamma \sum_{s' \in S} p(s' | s, a) \max_{a'} \tilde{q}_{\text{prev}}(s', a') \right) \\
\quad \Delta \leftarrow \max (\Delta, |\tilde{q}(s, a) - \tilde{q}_{\text{prev}}(s, a)|) \\
\text{end} \\
\text{end} \\
\text{until } \Delta < \delta
\]

output: Deterministic policy $\tilde{\pi}(s) \in \arg\max_a \tilde{q}(s, a)$

- Same complexity but more storage than with state value function...  
- but will be useful later!
Outline

1. Prediction and Bellman Equation
2. Prediction by Dynamic Programming and Contraction
3. Planning, Optimal Policies and Bellman Equation
4. Linear Programming
5. Planning by Value Iteration
6. Planning by Policy Iteration
7. Optimization Interpretation
8. Approximation and Stability
9. Generalized Policy Iteration
10. Infinite, Episodic and Average setting
11. References
Value Function vs Policy Point of View

\[ v, q \rightarrow \Pi \text{ or } \Pi \rightarrow v, q? \]

**Planning**

- Focus so far on value-function point of view!
- Heuristic: find a good approximation of the optimal value function and deduce a good policy.
- Can we work directly on the policy itself?

- For prediction, only the policy point of view makes sense!
∀s, \pi_+(s) \in \arg\max_a q_{\pi}(s, a) \implies \forall v_{\pi_+}(s) \geq v_{\pi}(s)

**Classical Policy Improvement Lemma**

- **Prop:** Given a policy \pi and its q value-function, one can obtain a better policy with the argmax operator.
- **Prop:** If no improvement is possible, it means that \pi is already optimal.
- **Proof:** Use \mathcal{T}_{\pi_+} v_{\pi} = \mathcal{T}^* v_{\pi} \geq \mathcal{T}^\pi v_{\pi} = v_{\pi} to prove \left(\mathcal{T}_{\pi_+}\right)^k v_{\pi} \geq v_{\pi} which implies the result by letting k goes to +\infty.

- Leads to a sequential improvement algorithm...
Policy Improvement Lemma

\[ \mathbb{E}[v_{\pi'}(S_0)] - \mathbb{E}[v_{\pi}(S_0)] = \sum_{k=0}^{\infty} \gamma^k \mathbb{E}_{\pi'} \left[ \sum_a \pi'(a|S_t) \left( q_\pi(S_t, a) - v_\pi(S_t) \right) \right] \]

\[ = \sum_{k=0}^{\infty} \gamma^k \mathbb{E}_{\pi'} \left[ \sum_a (\pi'(a|S_t) - \pi(a|S_t)) q_\pi(S_t, a) \right] \]

A Generic Improvement Lemma

- No assumptions on \( \pi \) and \( \pi' \)!
- Easy proof.
- Imply the previous lemma as \( \max_a Q_\pi(s, a) - v_\pi(s) \geq 0 \).
- Show that improvement choices are possible.
- Will prove to be useful later...
Policy Iteration

Discounted: Planning by Policy Iteration

**input:** MDP model $\langle (S,A,R), P \rangle$, and discount factor $\gamma$

**parameter:** Initial policy $\tilde{\pi}$

**repeat**

1. **repeat**
   2. **for** $s \in S$ **do**
   3. **for** $a \in A$ **do**
   4. $\tilde{\text{pol}}(s) \leftarrow \text{argmax } q_{\tilde{\pi}}(s, a)$
   5. **end**
   6. **end**

**output:** Deterministic policy $\tilde{\pi}$.

Some issues

- How to obtain $q_{\pi}$?
- When to stop?
Discounted: Planning by Policy Iteration

**input:** MDP model $\langle (S, A, R), P \rangle$, and discount factor $\gamma$

**parameter:** Initial policy $\tilde{\pi}$

**repeat**

- $stable \leftarrow 0$
- Compute $q_{\tilde{\pi}}$

  **for** $s \in S$ **do**

  - $old - action \leftarrow \tilde{\pi}(s)$
  - $\tilde{\pi}(s) \leftarrow \text{argmax } q_{\tilde{\pi}}(s, a)$
  - **if** $\tilde{\pi}(s) \neq old - action$ **then**
    - $stable \leftarrow 0$
  - **end**

**end**

**until** $stable == 1$

**output:** Deterministic policy $\tilde{\pi}$.

---

**Finite Setting**

- Finite set of action-states implies a finite set of policy.
- Convergence of the algorithm in finite time!
Policy Iteration

Convergence Rate

- Crude analysis:
  - Bound after $k$ steps of the algorithm
    \[
    \|v_{\pi_k} - v_\star\|_\infty \leq \gamma \|v_{\pi_k-1} - v_\star\|_\infty \leq \gamma^k \|v_{\pi_0} - v_\star\|_\infty
    \]
    \[
    \|v_{\pi_k} - v_\star\|_\infty \leq \frac{\gamma}{1 - \gamma} \|v_{\pi_k} - v_{\pi_{k-1}}\|_\infty
    \]
  - Not much better than value iteration but much higher complexity as $q_{\pi_k}$ is obtained by solving the Bellman equation!

- Much faster in practice...

- Clever analysis (Putterman):
  - Under some mild assumptions and provided $\|P_{\pi_k} - P_\star\| \leq K \|v_{\pi_k} - v_\star\|_\infty$ then
    \[
    \|v_{\pi_k} - v_\star\|_\infty \leq \frac{K \gamma}{1 - \gamma} \|v_{\pi_{k-1}} - v_\star\|_\infty
    \]
  - May explain the better convergence in practice!
Outline

1 Prediction and Bellman Equation
2 Prediction by Dynamic Programming and Contraction
3 Planning, Optimal Policies and Bellman Equation
4 Linear Programming
5 Planning by Value Iteration
6 Planning by Policy Iteration
7 Optimization Interpretation
8 Approximation and Stability
9 Generalized Policy Iteration
10 Infinite, Episodic and Average setting
11 References
Value Iteration: (Relaxed) First Order Method

**Value Iteration**

- **Iteration:**
  \[ v_k = \mathcal{T}^* v_{k-1} \]
  \[ = v_{k-1} + (\mathcal{T}^* - \text{Id}) v_{k-1} \]

- **Relaxation**
  \[ v_k = v_{k-1} - \alpha (\text{Id} - \mathcal{T}^*) v_{k-1} \]

  can be proved to converge for any \( \alpha < \frac{2}{1+\gamma} \).

- Can be interpreted as a first order method with pseudo-gradient \((\mathcal{T}^* - \text{Id}) v_{k-1}\).
- No function corresponding to this gradient!

- Is there a better choice for \( \alpha \) than \( \alpha = 1 \)?
- No as the resulting operator is a contraction of constant
  \[ |1 - \alpha| + \alpha \gamma \geq \gamma \]
Policy Iteration: Newton-Raphson Method

Policy Iteration

- Explicit iteration:

Solve \( \nu_{\pi_{k-1}} = \mathcal{T}^{\pi_k} \nu_{\pi_{k-1}} \)

Let \( \pi_k \) such that \( \mathcal{T}^{\pi_k} \nu_{\pi_{k-1}} = \mathcal{T}^* \nu_{\pi_{k-1}} \)

- Implicit iteration on \( \nu_{\pi_k} \):

\[
\begin{align*}
\nu_{\pi_k} &= (\text{Id} - \gamma P^{\pi_k})^{-1} r_{\pi_k} \\
&= (\text{Id} - \gamma P^{\pi_k})^{-1} (r_{\pi_k} + (\gamma P^{\pi_k} - \text{Id}) \nu_{\pi_{k-1}} + (\text{Id} - \gamma P^{\pi_k}) \nu_{\pi_{k-1}}) \\
&= \nu_{\pi_{k-1}} - (\text{Id} - \gamma P^{\pi_k})^{-1} (\text{Id} - \mathcal{T}^{\pi_k}) \nu_{\pi_{k-1}}
\end{align*}
\]

- Can be interpreted as a second order method with pseudo-gradient \( (\text{Id} - \mathcal{T}^{\pi_k}) \nu_{\pi_{k-1}} = (\text{Id} - \mathcal{T}^*) \nu_{\pi_{k-1}} \) and pseudo-Hessian \( (\text{Id} - \gamma P^{\pi_k}) \).

- Not a formal analysis but give a good insight on the better convergence of policy iteration.
Outline

1. Prediction and Bellman Equation
2. Prediction by Dynamic Programming and Contraction
3. Planning, Optimal Policies and Bellman Equation
4. Linear Programming
5. Planning by Value Iteration
6. Planning by Policy Iteration
7. Optimization Interpretation
8. Approximation and Stability
9. Generalized Policy Iteration
10. Infinite, Episodic and Average setting
11. References
Stability of Value and Policy Iteration

Ideal Value and Policy Iteration?

- Iterative algorithms.
- Convergence proofs assume perfect computation.
- What happens if we make a (small) error at each step?

- Particularly important for Policy Iteration in which one resolves a linear system at each step!
Value Iteration Stability

\[ v_k = \mathcal{T}^* v_{k-1} + \epsilon_{k-1} \]

\[ \Rightarrow \| v_k - v_* \|_{\infty} \leq \gamma^k \| v_0 - v_* \|_{\infty} + \frac{\max_{0 \leq k' < k} \| \epsilon_{k'} \|_{\infty}}{1 - \gamma} \]

\[ \Rightarrow \| v_{\pi_k} - v_* \|_{\infty} \leq \frac{2\gamma^{k+1}}{1 - \gamma} \| v_0 - v_* \|_{\infty} + \frac{2\gamma \max_{0 \leq k' < k} \| \epsilon_{k'} \|_{\infty}}{(1 - \gamma)^2} \]

**Stability with respect to the error**

- Proof relies on the contraction property of \( \mathcal{T}^* \) (hence similar results for \( \mathcal{T}^\pi \)).
- Error term \( \frac{\max_{0 \leq k' < k} \| \epsilon_{k'} \|_{\infty}}{1 - \gamma} \) can be replaced by \( \sum_{k'=0}^{k-1} \gamma^{k-k'} \| \epsilon_{k'} \|_{\infty} \)
- Convergence if \( \| \epsilon_k \|_{\infty} \) tends to 0.
- Remains in a neighborhood of the optimal solution if \( \| \epsilon_k \|_{\infty} \) is bounded.
Policy Iteration

\[ v_{k-1} = v_{\pi_{k-1}} + \epsilon_{k-1} \quad \text{and} \quad T^{\pi_k} v_{k-1} = T^* v_{k-1} \]

\[ \implies \| v_\pi_k - v_* \|_\infty \leq \gamma^k \| v_{\pi_0} - v_* \|_\infty + \frac{\gamma(2 - \gamma) \max_{0 \leq k' < k} \| \epsilon_{k'} \|_\infty}{(1 - \gamma)^2} \]

Stability with respect to the error

- Quite involved proof but crude results.
- Error term \( \max_{0 \leq k' < k} \| \epsilon_{k'} \|_\infty \) can be replaced by \( \sum_{k' = 0}^{k-1} \gamma^{k-k'} \| \epsilon_{k'} \|_\infty \)
- Convergence if \( \| \epsilon_k \|_\infty \) tends to 0.
- Remains in a neighborhood of the optimal solution if \( \| \epsilon_k \|_\infty \) is bounded.
- Policy Iteration only requires an approximate estimate of \( v_{\pi_{k-1}} \), for instance obtained by Bellman iteration...
Outline

1. Prediction and Bellman Equation
2. Prediction by Dynamic Programming and Contraction
3. Planning, Optimal Policies and Bellman Equation
4. Linear Programming
5. Planning by Value Iteration
6. Planning by Policy Iteration
7. Optimization Interpretation
8. Approximation and Stability
9. Generalized Policy Iteration
10. Infinite, Episodic and Average setting
11. References
Modified Policy Iteration

Discounted: Planning by Generalized Policy Iteration

**input:** MDP model \((S, A, R, P)\), and discount factor \(\gamma\)

**parameter:** Initial \(q\)

**repeat**

\[
\text{for } s \in S \text{ do} \quad \tilde{\pi}(s) \leftarrow \text{argmax } q(s, a) \\
\text{end repeat}
\]

\[
q_{\text{prev}} \rightarrow q
\]

**repeat**

\[
\text{for } (s, a) \in S \times A \text{ do} \\
q(s, a) \leftarrow r(s, a) + \gamma \sum_{s', a'} p(s'|s, a) \tilde{\pi}(a'|s) q_{\text{prev}}(s, a) \\
\text{end}
\]

**output:** Deterministic policy \(\tilde{\pi}\).

- Algorithm driven by \(q\).
- Flexibility in the number of prediction steps after each policy improvement steps.
- Special cases:
  - Large number: Policy Iteration with (small) error.
  - One: Value Iteration!
Generalized Policy Iteration

\[ T^{\pi_k} v_k = T^* v_k \quad \text{and} \quad v_{k+1} = (T^{\pi_k})^{m_k} v_k \]

\[ \implies \| v_{k+1} - v_* \|_\infty \leq \gamma \left( \frac{1 - \gamma^{m_k}}{1 - \gamma} \right) \| P^{\pi_k} - P^* \| + \gamma^{m_k} \| v_k - v_* \|_\infty \]

Convergence Results

- Quite technical proof.
- Valid only under the mild assumption \( T^* v_0 \geq v_0 \).
- Very fast decay provided \( \| P^{\pi_k} - P^* \| \) is small.
- No stability with arbitrary errors...
Generalized Policy Iteration

Two simultaneous interacting processes:
- One forcing the policy to correspond to the current value function (Policy Improvement)
- One trying to make the current value function coherent with the current policy (Policy Evaluation)

Several variations possible on the two processes.

- In GPI, the policy is driven by the value function.
- Typically, stabilizes only if one reaches the optimal value/policy pair.
Generalized Policy Iteration

**State Update Order**

**Discounted: Prediction by Value Iteration - State Update Order**

**input:** MDP model \( \langle (S, A, R), P \rangle \), discount factor \( \gamma \), and stationary policy \( \pi \)

**init:** \( \tilde{v}(s) \forall s \in S \)

**repeat**

\[
\tilde{v}_{\text{prev}} \leftarrow \tilde{v} \\
\text{for } s \in S' \subset S \text{ do} \\
\tilde{v}(s) \leftarrow \sum_{a \in A} \pi(a|s) \left( r(s, a) + \gamma \sum_{s' \in S} p(s'|s, a) \tilde{v}_{\text{prev}}(s') \right)
\]

**end**

**output:** Value function \( \tilde{v} \)

**Classical strategies**

- \( S' = S \): classical iteration
- \( S' = \{s\} \): Gauss-Seidel
- \( S' = \{s, |T^\pi \tilde{v}(s) - \tilde{v}(s)| > \epsilon\} \): Prioritized sweeping

- Converges provided all states are visited infinitely often...
- Gain in term of storage or focus on most interesting states...
Policy Improvement Variation

Greedy: \( \pi(s) \in \arg\max_a q(s, a) \iff \pi(\cdot|s) \in \arg\max_{\tilde{\pi}} \sum_a \tilde{\pi}(a)q(s, a) \)

Restricted: \( \pi(\cdot|s) \in \arg\max_{\tilde{\pi} \in \tilde{\Pi}_\epsilon} \sum_a \tilde{\pi}(a)q(s, a) \)

Regularized: \( \pi(\cdot|s) \in \arg\max_{\tilde{\pi}} \sum_a \tilde{\pi}(a)q(s, a) + \epsilon P(\tilde{\pi}) \)

Classical Variations

- \( \epsilon \)-greedy: Restrict \( \tilde{\pi} \) to the set of policy s.t. \( \tilde{\pi}(a) \geq \epsilon \)
  - Explicit solution: \( \pi(a|s) = \epsilon + (1 - \epsilon) \arg\max_a q(s, a) \)
  - Policy improvement property if \( \epsilon \) decreases.

- Soft-max: Regularize by \( \epsilon H(\tilde{\pi}) \) where \( H \) is the entropy.
  - Explicit solution: \( \pi(a|s) \propto \exp(q(s, a)/\epsilon) \)
  - No classical policy improvement...

- Tends to greedy when \( \epsilon \) goes to 0.
- Will proved to be interesting later...
Outline

1. Prediction and Bellman Equation
2. Prediction by Dynamic Programming and Contraction
3. Planning, Optimal Policies and Bellman Equation
4. Linear Programming
5. Planning by Value Iteration
6. Planning by Policy Iteration
7. Optimization Interpretation
8. Approximation and Stability
9. Generalized Policy Iteration
10. Infinite, Episodic and Average setting
11. References
Infinite Setting

- No issue with the rewards as only the expectation is used.
- All the theory remains valid if the states are countable, but there is an issue in the algorithms as we need to store/update an infinite number of states.
- The proof of existence of an optimal policy requires the max to be attained, which cannot be ensured in an infinite (even countable setting).

Some results...

**Thm:** If $S$ is countable, there exists an $\epsilon$-optimal (stationary) policy for any $\epsilon > 0$.

**Thm:** If $S$ is a Polish space (completely metrizable topological space),
- there exists a $(P, \epsilon)$-optimal (stationary policy) for any $\epsilon > 0$.
- if each $A_s$ is countable, there exists an $\epsilon$-optimal (stationary) policy for any $\epsilon > 0$.
- if each $A_s$ is finite, there exists an optimal (stationary) policy.
- if each $A_s$ is a compact metric space, $r(s, a)$ is a bounded u.s.c. function on $A_s$ and $p(B|s, a)$ is continuous in $a$ for each Borel subset $B$ and any $s$, there exist an optimal (stationary) policy.

- Mainly technical difficulties...
Total Reward

\[ v_\Pi(s) = \mathbb{E}_\Pi \left[ \sum_{t'=1}^{+\infty} R_{t+1} \middle| S_0 = s \right] = \mathbb{E}_\Pi \left[ \sum_{t'=1}^{+\infty} \max(0, R_{t+1}) \middle| S_t = s \right] - \mathbb{E}_\Pi \left[ \sum_{t'=t+1}^{+\infty} \max(0, -R_{t+1}) \middle| S_t = s \right] \]

- Total reward not necessarily well defined!
- Need to **assume** this is the case!

**Classical Assumptions**

- Episodic model: \( \forall \Pi, s, \mathbb{E}_\Pi \left[ \min_{t, \forall t' \geq t, \forall R_{t'}} S_0 = s \right] < +\infty \)
- More general assumption: \( \forall \Pi, s \) either \( v_{+,\Pi}(s) \) or \( v_\Pi(s) \) is finite.
Bellman Operator and Optimality Equation

\[ \sup_{\Pi} v_\Pi(s) = v_*(s) = \max_a r(s, a) + \sum_{s'} p(s'|s, a) v_*(s') \]

- Similar to the discounted setting as:
  - We can focus on Markovian policy.
  - The optimal value \( v_* \) satisfies the Bellman optimality equation.

**But...**

- \( \mathcal{T}^* \) is not a contraction and thus there may be several solution of the equation.
- If \( \pi \) is such that \( \mathcal{T}^\pi v_* = \mathcal{T}^* v_* \), we need to assume that \( \limsup (P^\pi)^n v_*(s) \leq 0 \) to prove that \( \Pi = (\pi, \pi, \ldots) \) is optimal.
- There may not exists an optimal policy!

- Existence of optimal policies in the finite state-action setting by defining the total reward to the limit of discounted setting when \( \gamma \to 1 \) and using the finiteness of the policy set...
### Positive Bounded Models

- \( \forall \Pi, s, \; \nu_{+},\Pi(s) < \infty \)
- \( \forall s, \exists a, \; r(s, a) \geq 0 \)

Often stronger assumption: \( r(s, a) \geq 0 \).

Any discounted model can be put in this framework by adding an absorbing state reached at random at each step with probability \( 1 - \gamma \).

### Negative Models

- \( \forall \Pi, s, \; \nu_{+},\Pi(s) = 0 \) and \( \nu_{-},\Pi(s) < \infty \)
- There exists a policy \( \Pi \) such that \( \forall s, \; \nu_{\Pi}(s) > -\infty \)

- Maximization of \( \nu_{\Pi} \) amounts to the minization of \( \nu_{-},\Pi \) and the negative reward can be interpreted as the opposite of costs.
- Stochastic Shortest Path within this framework.
### Positive Bounded and Negative Models Results

<table>
<thead>
<tr>
<th>Result</th>
<th>Positive Bounded Models</th>
<th>Negative Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimality equation</td>
<td>$v^<em>$ is a minimal solution within $v \leq T^</em> v$</td>
<td>$v^*$ is a maximal solution within $v \geq T v$</td>
</tr>
<tr>
<td>$T^\pi v_* = T^* v_* \Rightarrow \pi$ optimal</td>
<td>Only if $\lim \sup (P^\pi)^n v_*(s) = 0$</td>
<td>Always</td>
</tr>
<tr>
<td>Existence of optimal stationary policy</td>
<td>$S$ and $A$ finite or existence of optimal policy and $r \geq 0$</td>
<td>$A_s$ finite or $A_s$ compact, $r$ and $p$ continuous with respect to $a$.</td>
</tr>
<tr>
<td>Existence of stationary $\epsilon$-optimal policy</td>
<td>If $v^*$ is bounded</td>
<td>Not always (Always for non stationary policy)</td>
</tr>
<tr>
<td>Value Iteration converges</td>
<td>$0 \leq v_0 \leq v_*$</td>
<td>$0 \geq v_0 \geq v_<em>$ and $A_s$ finite or $S$ finite if $v_</em> &gt; -\infty$</td>
</tr>
<tr>
<td>Policy Iteration converges</td>
<td>Yes</td>
<td>Not always</td>
</tr>
<tr>
<td>Modified Policy Iteration converges</td>
<td>$0 \leq v_0 \leq v_<em>$ and $v_0 \leq T^</em> v_0$</td>
<td>Not always</td>
</tr>
<tr>
<td>Solution by linear programming</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
Average Return

\[
\overline{\nu}_\Pi(s) = \lim_{T \to \infty} \frac{1}{T} \nu_{T,\Pi}(s) = \lim_{T \to \infty} \frac{1}{T} \mathbb{E}_\Pi \left[ \sum_{t=1}^{T} R_t \middle| S_0 = s \right]
\]

\[
\longrightarrow \overline{\nu}_+,\Pi(s) = \limsup_{T \to \infty} \frac{1}{T} \nu_{T,\Pi}(s)
\]

\[
\overline{\nu}_-,\Pi(s) = \liminf_{T \to \infty} \frac{1}{T} \nu_{T,\Pi}(s)
\]

**Average Return(s)**

- Limit \( \overline{\nu}_\Pi \) may not be defined!

- **Prop:** \( \overline{\nu}_\Pi \) is well defined if \( \Pi \) is stationary and \( \frac{1}{T} \sum_{t=1}^{T} (P^\pi)^{t-1} \) tends to a stochastic matrix.

- Limits \( \overline{\nu}_+,\Pi \) and \( \overline{\nu}_-,\Pi \) always defined!
Average Returns and Optimality

\[ \bar{v}_{+,*}(s) = \sup_{\Pi} \bar{v}_{+,\Pi}(s) \quad \text{and} \quad \bar{v}_{-,*}(s) = \sup_{\Pi} \bar{v}_{-,\Pi}(s) \]

**Optimality of \( \Pi_* \)**

- **Average optimal:**
  \[ \bar{v}_{-,\Pi_*} \geq \bar{v}_{+,*}(s) \]

- **Lim-sup average optimal (best case analysis):**
  \[ \bar{v}_{+,\Pi_*} \geq \bar{v}_{+,*}(s) \]

- **Lim-inf average optimal (worst case analysis):**
  \[ \bar{v}_{-,\Pi_*} \geq \bar{v}_{-,*}(s) \]

- More complex setting!
- Let’s start with Prediction...
Infinite, Episodic and Average setting

\[ \overline{v}_\pi(s) = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} P^{t-1}_\pi r_\pi = \left( \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} P^{t-1}_\pi \right) r_\pi = P^\infty_\pi r_\pi \]

Stochastic Matrix \( P^\infty_\pi \)

- Measures the average amount of time spend on a state \( s' \) starting from state \( s \) at \( t = 0 \) when using policy \( \pi \).
- Structure linked to the properties of the resulting Markov chain:
  - If aperiodic, \( P^\infty_\pi = \lim_T P^T_\pi \) i.e. \( P^\infty_\pi \) is close to the probability of reaching \( s' \) from \( s \) at any large \( T \).
  - If unichain, then \( P^\infty_\pi \) has identical rows and corresponds to the stationary distribution.
  - If multichain, then \( P^\infty_\pi \) has a diagonal block structure with rows equal withing each block corresponding to the stationary distribution in each chain.

- Implies that \( \overline{v}_\pi(s) = \overline{v}_\pi(s') \) in the Markov process is unichain.
- Limit \( P^\infty_\pi \) may be hard to compute...
Infinite, Episodic and Average setting

Average Reward and Relative Value Functions

\[ U_\pi(s) = \mathbb{E}_\pi \left[ \sum_{t=1}^{\infty} (R_t - \bar{v}_\pi(S_t)) \middle| S_0 = s \right] \]

\[ \iff \quad U_\pi = \left( \text{Id} - P_\pi + P_\pi^\infty \right)^{-1} \left( \text{Id} - P_\pi^\infty \right) r_\pi \]

Link between \( U_\pi \) and \( \bar{v}_\pi \)

- \((\text{Id} - P_\pi)\bar{v}_\pi = 0\)
- \(\bar{v}_\pi + (I - P_\pi)U_\pi = r_\pi\)

Characterization by a system

- If \((\text{Id} - P_\pi)\bar{v} = 0\) and \(\bar{v} + (I - P_\pi)U = r_\pi\) then
  - \(\bar{v} = \bar{v}_\pi\),
  - \(U = U_\pi + u\) with \((I - P_\pi)u = 0\),
  - If \(P_\pi^\infty U = 0\) then \(u = 0\).

- Prediction possible by solving this system as we do not need \(U_\pi\).
**Optimality Equations**

\[
\bar{v}(s) = \max_a \sum_{s'} p(s'|s, a)\bar{v}(s')
\]

\[
U(s) + \bar{v}(s) = \max_{a \in B_s} r(s, a) + \sum_{s'} p(s'|s, a)U(s) \text{ with } B_s = \{ a | \sum_{s'} p(s'|s, a)\bar{v}(s') = \bar{v}(s) \}
\]

\[
\pi_* (s) \in \arg\max_{a \in B_s} r(s, a) + \sum_{s'} p(s'|s, a)U(s)
\]

**Existence**

- If there is a solution \((\bar{v}, U)\) of the system then \(\bar{v} = \bar{v}_*\) and \(\pi_*\) is an optimal policy.
- There may exist other optimal policies not satisfying the \(\arg\max\) property.
- There may not exist solutions to the system.

- Associated relative value iteration and modified policy iteration can be defined.
- Convergence under strong assumptions…
Outline

1 Prediction and Bellman Equation
2 Prediction by Dynamic Programming and Contraction
3 Planning, Optimal Policies and Bellman Equation
4 Linear Programming
5 Planning by Value Iteration
6 Planning by Policy Iteration
7 Optimization Interpretation
8 Approximation and Stability
9 Generalized Policy Iteration
10 Infinite, Episodic and Average setting
11 References
References

R. Sutton and A. Barto. 
*Reinforcement Learning, an Introduction (2nd ed.)*
MIT Press, 2018

O. Sigaud and O. Buffet. 
*Markov Decision Processes in Artificial Intelligence.*
Wiley, 2010

M. Puterman. 
Wiley, 2005

D. Bertsekas and J. Tsitsiklis. 
*Neuro-Dynamic Programming.*
Athena Scientific, 1996

W. Powell. 
Wiley, 2022

S. Meyn. 
*Control Systems and Reinforcement Learning.*
Cambridge University Press, 2022

V. Borkar. 
*Stochastic Approximation: A Dynamical Systems Viewpoint.*
Springer, 2008

T. Lattimore and Cs. Szepesvári. 
*Bandit Algorithms.*
Cambridge University Press, 2020
Licence and Contributors

Creative Commons Attribution-ShareAlike (CC BY-SA 4.0)

- You are free to:
  - **Share**: copy and redistribute the material in any medium or format
  - **Adapt**: remix, transform, and build upon the material for any purpose, even commercially.
- Under the following terms:
  - **Attribution**: You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
  - **ShareAlike**: If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
  - **No additional restrictions**: You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Contributors

- **Main contributor**: E. Le Pennec