Dynamic Programming: Algorithms

John Stachurski

June 2024

Topics

DP algorithms:

- Value function iteration (VFI)
- Howard policy iteration (OPI)
- Optimistic policy iteration (HPI)

What convergence properties?

How do they interact with parallelization?

Reference / proofs:

https://dp.quantecon.org/

Example: optimal savings

Wealth evolves according to

$$w_{t+1} = Rw_t - c_t + y_t$$

- y is labor income
- w is wealth
- R is the gross rate of return on assets

Bellman equation:

$$v(w, y) = \max_{0 \leqslant w' \leqslant w} \left\{ u(Rw + y - w') + \beta \sum_{y'} v(w', y') Q(y, y') \right\}$$

A generalization:

$$v(x) = \max_{a \in \Gamma(x)} \left\{ r(x,a) + \beta \sum_{x' \in \mathsf{X}} v(x') P(x,a,x') \right\}$$

- $x \in \mathsf{X}$ is the **state**
- $a \in \mathsf{A}$ is the **action**
- $\Gamma(x) =$ actions available in state x

Policies

A **feasible policy** is a map σ from X to A such that

$$\sigma(x) \in \Gamma(x) \text{ for all } x \in \mathsf{X}$$

- respond to X_t with action $A_t = \sigma(X_t)$ at all $t \geqslant 0$

Let

$$\Sigma = \text{the set of all feasible policies}$$

Let $v_\sigma(x)$ = lifetime value of policy σ, starting from x

The function v_σ satisfies

$$v_\sigma(x) = r(x, \sigma(x)) + \beta \sum_{x'} v_\sigma(x') P(x, \sigma(x), x')$$

Letting

- $P_\sigma(x, x') = P(x, \sigma(x), x')$ = Markov dynamics given σ and
- $r_\sigma(x) = r(x, \sigma(x))$ = rewards at x given σ

we can write this as the vector equation

$$v_\sigma = r_\sigma + \beta P_\sigma v_\sigma$$

How to solve

$$v_\sigma = r_\sigma + \beta P_\sigma v_\sigma$$

for v_σ?

Option 1: Use linear algebra to obtain

$$v_\sigma = (I - \beta P_\sigma)^{-1} r_\sigma$$

Option 2: Define the **policy operator** corresponding to σ is

$$(T_\sigma v)(x) = r(x, \sigma(x)) + \beta \sum_{x' \in \mathsf{X}} v(x') P(x, \sigma(x), x')$$

In vector notation,

$$T_\sigma v = r_\sigma + \beta P_\sigma v$$

Fact. T_σ is a contraction map on $\mathbb{R}^n$

Proof: Follows from

$$|T_\sigma v - T_\sigma w| \leqslant \beta |P_\sigma v - P_\sigma w|$$

Fact. v_σ is the unique fixed point of T_σ in $\mathbb{R}^n$

Proof: Since $\beta < 1$, we have

$$v = T_\sigma v \iff v = r_\sigma + \beta P_\sigma v$$

$$\iff v = (I - \beta P_\sigma)^{-1} r_\sigma$$

$$\iff v = v_\sigma$$

Hence

$$v \text{ is a fixed point of } T_\sigma \iff v = v_\sigma$$

Fact. v_σ is the unique fixed point of T_σ in $\mathbb{R}^n$

Proof: Since $\beta < 1$, we have

$$
\begin{aligned}
v = T_\sigma \, v &\iff v = r_\sigma + \beta P_\sigma v \\
&\iff v = (I - \beta P_\sigma)^{-1} \, r_\sigma \\
&\iff v = v_\sigma
\end{aligned}
$$

Hence

$$v \text{ is a fixed point of } T_\sigma \quad \iff \quad v = v_\sigma$$

Greedy Policies

Fix $v \in \mathbb{R}^n$

A policy σ is called v-**greedy** if

$$\sigma(x) \in \operatorname*{argmax}_{a \in \Gamma(x)} \left\{ r(x,a) + \beta \sum_{x'} v(x') P(x,a,x') \right\}$$

for all $x \in \mathsf{X}$

Ex. Prove: at least one v-greedy policy exists in Σ

The **Bellman operator** is defined by

$$(Tv)(x) = \max_{a \in \Gamma(x)} \left\{ r(x,a) + \beta \sum_{x'} v(x') P(x,a,x') \right\}$$

By construction,

$$Tv = v \iff v \text{ satisfies the Bellman equation}$$

Optimality

The **value function** is defined by

$$v^*(x) := \max_{\sigma \in \Sigma} v_\sigma(x) \qquad (x \in \mathsf{X})$$

A policy $\sigma \in \Sigma$ is called **optimal** if

$$v_\sigma = v^*$$

Standard theory (Bellman, Denardo, Blackwell)

Theorem. For the DP model described above,

1. v^* is the unique fixed point of T in $\mathbb{R}^n$
2. A feasible policy is optimal if and only it is v^*-greedy
3. At least one optimal policy exists

Algorithms

1. Value function iteration (HPI)
2. Howard policy iteration (HPI)
3. Optimistic policy iteration (OPI)

Algorithm 1: VFI for MDPs

input $v_0 \in \mathbb{R}^n$

input τ

$\varepsilon \leftarrow \tau + 1$

$k \leftarrow 0$

while $\varepsilon > \tau$ **do**

  $v_{k+1} \leftarrow T v_k$

  $\varepsilon \leftarrow \|v_k - v_{k+1}\|_\infty$

  $k \leftarrow k + 1$

end

Compute a v_k-greedy policy σ

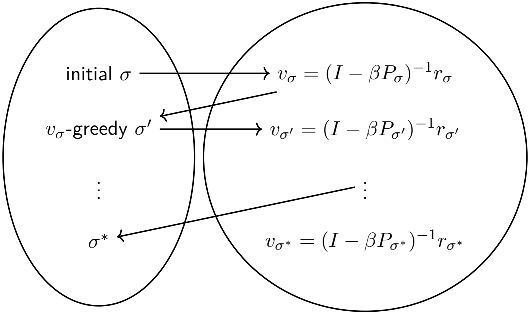
return σ

VFI is

- easy to understand
- easy to implement
- globally convergent

But the convergence rate is only linear

Howard Policy Iteration



Iterates between computing the value of a given policy and computing the greedy policy associated with that value

Algorithm 2: Howard policy iteration for MDPs

input $\sigma \in \Sigma$

$v_0 \leftarrow v_\sigma$ and $k \leftarrow 0$

repeat

  $\sigma_k \leftarrow$ a v_k-greedy policy

  $v_{k+1} \leftarrow (I - \beta P_{\sigma_k})^{-1} r_{\sigma_k}$

  if $v_{k+1} = v_k$ **then** break

  $k \leftarrow k + 1$

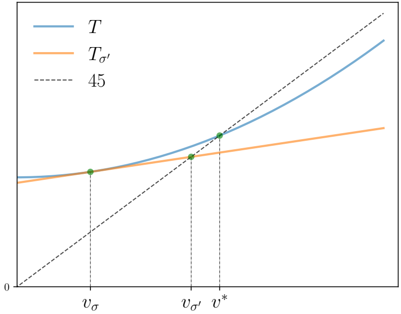
return σ_k

Proposition. HPI returns an exact optimal policy in a finite number of steps

Also, rate of convergence is faster than VFI

In fact HPI $=$ gradient-based Newton iteration on T

- Implies a quadratic rate of convergence
- Details are in `https://dp.quantecon.org`



- σ' is v_σ-greedy if $T_{\sigma'} v_\sigma = T v_\sigma$
- $v_{\sigma'}$ is the fixed point of $T_{\sigma'}$

Optimistic Policy Iteration

OPI is a "convex combination" of VFI and HPI

Similar to HPI except that

- HPI takes current σ and obtains v_σ
- OPI takes current σ and iterates m times with T_σ

Recall that, for any $v \in \mathbb{R}^n$, we have $T_\sigma^m v \to v_\sigma$ as $m \to \infty$

Hence OPI replaces v_σ with an approximation

Algorithm 3: Optimistic policy iteration for MDPs

input $v_0 \in \mathbb{R}^n$

input $\tau > 0$ and $m \in \mathbb{N}$, a step size

$k \leftarrow 0$

$\varepsilon \leftarrow \tau + 1$

while $\varepsilon > \tau$ **do**

  $\sigma_k \leftarrow$ a v_k-greedy policy

  $v_{k+1} \leftarrow T_{\sigma_k}^m v_k$

  $\varepsilon \leftarrow \|v_k - v_{k+1}\|_\infty$

  $k \leftarrow k + 1$

end

return σ_k

Proposition. For all values of m we have $v_k \to v^*$

It's easy to show that OPI = VFI when $m = 1$

On the other hand, m is large, OPI is similar to HPI

- because $\lim_{m\to\infty} T_{\sigma_k}^m v_k = v_{\sigma_k}$

Rules of thumb:

- parallelization favors HPI – small number of expensive steps
- OFI is simple and dominates VFI for many values of m
- VFI works well when β is small and optimization is cheap