Residual-Conditioned Policy Iteration for Markov Games and Robust Markov Decision Processes

Jefferson Huang, PhD

Assistant Professor
Department of Operations Research
Naval Postgraduate School

Management Science Seminar

Lancaster University

Lancaster, UK
14 January, 2026

Summary

1. (Two-player zero-sum) **Markov games** are Markov decision processes with two diametrically opposed players.

2. Pollatschek & Avi-Itzhak (1969) proposed using **Newton's method** to solve Markov games.
   - Usually works and is fast in practice, but there are (simple) instances where it doesn't work.

3. Filar & Tolwinski (1991) proposed using **backtracking line search** instead.
   - Their convergence proof contains a gap; we construct an instance where convergence fails.

4. We propose **residual-conditioned policy iteration (RCPI)**, which matches the empirical performance of Newton's method while being provably convergent.*

*This talk is based on the following paper: Badger, Huang, and Petrik, *Fast Policy Iteration in Markov Games and Robust MDPs*, Proceedings of the 40th Annual AAAI Conference on Artificial Intelligence, 2026.

Markov Games

Definition

A *two-player zero-sum* **Markov game** is a Markov decision process where two players jointly control the one-step payoffs and transition probabilities.

- **Player 1** wants to *maximize* payoffs.
- **Player 2** wants to *minimize* payoffs.

On each time step $t = 0, 1, \ldots$:

1. Both players observe the current **state** $S_t \in \mathcal{S}$.
2. Based on the current state,
   - 2.1 Player 1 selects a feasible **action** $A_t \in \mathcal{A}(S_t)$, and
   - 2.2 *simultaneously*, Player 2 selects a feasible **action** $B_t \in \mathcal{B}(S_t)$.
3. A **one-step payoff** $r(S_t, A_t, B_t)$ is paid to Player 1 by Player 2.
4. The next state is S_{t+1} according to the **transition probabilities** $p(\cdot|S_t, A_t, B_t)$.

Total Discounted Payoffs

- Player 1's "decision variable" is a (possibly randomized) policy $\pi_1 \in \Pi_1$.
- Player 2's "decision variable" is a (possibly randomized) policy $\pi_2 \in \Pi_2$.

Given a **discount factor** $\gamma \in [0, 1)$, Player 1 wants to *maximize* the expected value of the total discounted payoff given the initial state $s \in \mathcal{S}$, i.e.,

$$v^{(\pi_1,\pi_2)}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r[S_t, \pi_1(A_t|S_t), \pi_2(B_t|S_t)] \;\middle|\; S_0 = s\right],$$

while Player 2 wants to *minimize* it.

Definition

A pair $(\pi_1^*, \pi_2^*) \in \Pi_1 \times \Pi_2$ of policies is **optimal** if they are Nash equilibrium policies for every starting state, in the sense that

$$v^{(\pi_1,\pi_2^*)}(s) \leqslant v^{(\pi_1^*,\pi_2^*)}(s) \leqslant v^{(\pi_1^*,\pi_2)}(s) \qquad \forall \;\; \pi_1 \in \Pi_1,\; \pi_2 \in \Pi_2,\; s \in \mathcal{S}.$$

The Value Function of the Markov Game

Shapley (1950) showed that when the state set $\mathcal{S}$ and action sets $\mathcal{A}(s), \mathcal{B}(s)$ are finite, there exists an optimal policy pair (π_1^*, π_2^*).

An optimal policy pair can be derived from the **value function**

$$v^*(s) = \max_{\pi_1 \in \Pi_1} \min_{\pi_2 \in \Pi_2} v^{(\pi_1, \pi_2)}(s), \quad s \in \mathcal{S}.$$

The value function is the *unique* solution to a nonlinear system of equations

$$v = \mathfrak{T}v$$

called the **optimality equations**.

- For each state $s \in \mathcal{S}$, $\mathfrak{T}v(s)$ is the value of the one-shot zero-sum game with payoff matrix $\mathbf{A}(s, v) = [\mathbf{A}_{a,b}(s, v)]$ with entries

$$\mathbf{A}_{a,b}(s, v) = r(s, a, b) + \gamma \sum_{s' \in \mathcal{S}} v(s') p(s'|s, a, b)$$

Finding Optimal Strategy Pairs

Definition

Given a $v \in \mathbb{R}^{\mathcal{S}}$, a policy pair (π_1^*, π_2^*) is **greedy** with respect to v if for every state $s \in \mathcal{S}$, the probability distributions $\pi_1^*(\cdot|s)$ and $\pi_2^*(\cdot|s)$ are Nash equilibrium strategies for the zero-sum matrix game $\mathbf{A}(s, v)$.

- $\mathfrak{G}(v) =$ set of all policy pairs that are greedy with respect to v.

- Shapley (1950) showed that any policy pair (π_1^*, π_2^*) that is greedy with respect to the value function v^* is optimal.

Computing the Value Function

Since the value function $v^* \in \mathbb{R}^{\mathcal{S}}$ is the unique solution to the optimality equations

$$v = \mathfrak{T} v$$

the problem of finding v^* can be solved by finding the (unique) solution to the unconstrained optimization problem

$$\underset{v \in \mathbb{R}^{\mathcal{S}}}{\text{minimize}} \quad f(v) = \frac{1}{2} \|\mathfrak{T} v - v\|_2^2 = \frac{1}{2} (\mathfrak{T} v - v)^\intercal (\mathfrak{T} v - v)$$

- A natural approach is to apply a **line search** method (e.g., Newton's method).

Line Search Methods

Recall:

We want to solve

$$\underset{v \in \mathbb{R}^S}{\text{minimize}} \quad f(v)$$

Starting from an initial guess $v^0 \in \mathbb{R}^S$, a **line search** method computes (hopefully) successively better solution estimates $v^1, v^2, \ldots$ using a formula of the form

$$v^{k+1} = v^k + \alpha_k d^k$$

where:

- $d^k \in \mathbb{R}^S$ is the **search direction**, and
- $\alpha_k \in \mathbb{R}$ is the **step size**.

When f is differentiable, the search directions usually have the form

$$d^k = -\mathbf{B}_k^{-1} \nabla f(v^k), \;\; k = 0, 1, \ldots,$$

where each matrix $\mathbf{B}_k$ is positive definite.

- **Newton's Method:** $\alpha_k = 1$ for all k.
- **Backtracking Line Search:** Use "Armijo's rule" to pick step sizes.

Convergence can be guaranteed when f is *continuously differentiable*; see e.g., Chapter 3 in Nocedal & Wright (2006).

Search Directions for Markov Games

For Markov games, the search direction given the current guess $v^k \in \mathbb{R}^{\mathcal{S}}$ is

$$d^k = -(\gamma \mathbf{P}^{(\pi_1,\pi_2)} - \mathbf{I})^{-1}(\mathfrak{T}v^k - v^k) \qquad \text{for some } (\pi_1, \pi_2) \in \mathfrak{G}(v^k),$$

where $\mathbf{P}^{(\pi_1,\pi_2)}$ is the transition matrix of the Markov chain on the state set $\mathcal{S}$ obtained when Players 1 and 2 respectively follow policies π_1 and π_2.

Newton's Method May Not Converge (van der Wal, 1978)

Example 2.1. Consider the Markov game (Fig. 1) with two states, both players having two actions in state 1 and only one in state 2. The notation reads as follows. If, in state 1, P_1 takes action 1 and P_2 takes action 2, then P_1 receives 6 from P_2, the system remains in state 1 with probability 1/3 and moves to state 2 with probability 2/3, etc. So, state 2 is absorbing. The discount factor used here is 3/4.

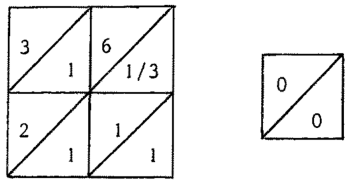


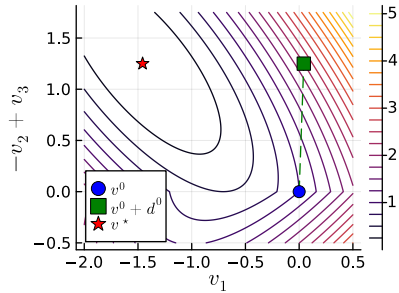
Fig. 1. Two-state Markov game.

Backtracking Line Search May Fail (Badger, H., and Petrik, 2026)

- $\mathcal{S} = \{s_1, s_2, s_3\}$
- $\mathcal{A}(1) = \mathcal{A}(2) = \mathcal{A}(3) = \{a_1\}$
- $\mathcal{B}(1) = \{b_1, b_2\}$, $\mathcal{B}(2) = \mathcal{B}(3) = \{b_1\}$
- In state s_1,
  - action b_1 incurs a one-step reward of $-\sqrt{2}/2$, and the next state is s_3 with probability 1, and
  - action b_2 incurs a one-step reward of $-\sqrt{2}/2$, and the next state is s_2 with probability 1.
- States s_2 and s_3 are absorbing states, with respective one-step rewards $-1/2$ and $1/2$.

For a discount factor $\gamma = 0.6$, backtracking line search will "backtrack forever" when the initial guess is $v_0 \equiv 0$.

Visualization from the paper:



Residual-Conditioned Policy Iteration (RCPI)

Newton's method may diverge because the iterates v^k may not provide a big enough reduction in the objective function $f(v) = \frac{1}{2}\|\mathfrak{T}v - v\|_2^2$.

Solution: On each iteration, make sure that the 2-norm of the **Bellman residual** $\mathfrak{T}v - v$ is reduced enough in order to guarantee convergence to the value function v^*.

See Badger, H., and Petrik (2026) for:

- the details of RCPI, and
- numerical comparisons to Newton's method (and others!) on randomly generated Markov games and Markov games modeling the robust control of "gridworld", gambler's ruin, and inventory control environments.

Recap

1. (Two-player zero-sum) **Markov games** are Markov decision processes with two diametrically opposed players.

2. Pollatschek & Avi-Itzhak (1969) proposed using **Newton's method** to solve Markov games.
   - Usually works and is fast in practice, but there are (simple) instances where it doesn't work.

3. Filar & Tolwinski (1991) proposed using **backtracking line search** instead.
   - Their convergence proof contains a gap; we construct an instance where convergence fails.

4. We propose **residual-conditioned policy iteration (RCPI)**, which matches the empirical performance of Newton's method while being provably convergent.[†]

[†] This talk is based on the following paper: Badger, Huang, and Petrik, *Fast Policy Iteration in Markov Games and Robust MDPs*, Proceedings of the 40th Annual AAAI Conference on Artificial Intelligence, 2026.