NEW RESULTS

IN TEMPORAL DIFFERENCE (TD) METHODS

Dimitri P. Bertsekas, MIT

Based on the Papers

- D. P. Bertsekas, V. Borkar, and A. Nedic, "Improved Temporal Difference Methods with Linear Function Approximation," in "Learning and Approximate Dynamic Programming", by A. Barto, W. Powell, J. Si, (Eds.), IEEE Press, 2004.

- A. Nedic and D. P. Bertsekas, "Least-Squares Policy Evaluation Algorithms with Linear Function Approximation," J. of Discrete Event Systems, Vol. 13, 2003.

- D. P. Bertsekas and S. Ioffe, "Temporal Differences-Based Policy Iteration and Applications in Neuro-Dynamic Programming," Report LIDS-P-2349, MIT, 1996; also in Neuro-Dynamic Programming book.

- H. Yu and D. P. Bertsekas, "Convergence Results for Some Temporal Difference Methods Based on Least Squares," Report LIDS 2697, MIT, 2006.

SUBJECT

- Approximate Dynamic Programming for large and intractable problems.

- Context: Discounted infinite horizon Markov Decision Problems.

- Construct simulation-based approximations to the cost-to-go function of a single stationary policy.

- Use the approximations in a policy iteration scheme.

- Extensions: stochastic shortest path problems (Bertsekas and Ioffe, 1996) and average cost problems (Yu and Bertsekas, 2006).

OUTLINE

- We discuss the Least Squares Policy Evaluation Method (λ-LSPE) a linear function approximation method for the cost function of a stationary policy (original proposal: Bertsekas and Ioffe, 1996)
  - It uses simulation and temporal differences (TD)
  - Uses a Kalman filter-like recursion.
  - It bears to TD(λ) the same type of relation as the Kalman filter/Gauss-Newton algorithm bears to the gradient method.
  - It converges with unit stepsize (Bertsekas, Borkar, Nedic, 2004)
  - It converges dramatically faster than TD(λ).
- Explain the underlying contraction mapping and its stochastic implementation via simulation.
- Explain the connection to value iteration with function approximation.

BACKGROUND ON DP

- Consider a classical discounted problem.
- A system with states $i = 1, \ldots, n$, and transition probabilities $p_{ij}(u)$ that depend on the control u.
- Cost of a policy $\pi = \{\mu_0, \mu_1, \ldots\}$

$$J_\pi(x_0) = \lim_{N \to \infty} E \left\{ \sum_{k=0}^{N-1} \alpha^k g\big(i_k, \mu_k(i_k), i_{k+1}\big) \right\}$$

where $\alpha < 1$ (discount factor).

- Shorthand notation for the DP mapping

$$(TJ)(i) = \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u) \big[g(i,u,j) + \alpha J(j) \big], \quad \forall\ i$$

- Similarly, define for any stationary policy μ

$$(T_\mu J)(i) = \sum_{j=1}^{n} p_{ij}\big(\mu(i)\big) \big[g(i,\mu(i),j) + \alpha J(j) \big], \quad \forall\ i$$

VALUE AND POLICY ITERATION

- Bellman's equation for the optimal cost J^* and the cost of a policy μ:

$$J^* = TJ^*, \qquad J_\mu = T_\mu J_\mu$$

- **Value iteration:** Start with any bounded J, and repeatedly apply T:

$$J^*(i) = \lim_{k\to\infty} (T^k J)(i)$$

Converges to J^* because T is a contraction.

- Similarly, we can compute T_μ by value iteration:

$$J_\mu(i) = \lim_{k\to\infty} (T_\mu^k J)(i)$$

- **Policy iteration:** Given μ^k,
  - *Policy evaluation:* Find J_{μ^k} by value iteration (or by solving the equation $J_{\mu^k} = T_{\mu^k} J_{\mu^k}$)
  - *Policy improvement:* Find μ^{k+1} such that

$$T_{\mu^{k+1}} J_{\mu^k} = TJ_{\mu^k}$$

VALUE ITERATION W/ FUNCTION APPROXIMATION

- Suppose we use linear approximations $\tilde{J}(i,r) = \phi(i)'r$, $i = 1,\ldots,n$, or

$$\tilde{J} = \Phi r \qquad \text{feature subspace approximation}$$

where $r = (r_1,\ldots,r_s)$ is a parameter vector, and Φ is a full rank $n \times s$ given matrix.

- **Approximate value iteration method:** Start with initial guess r_0 and iterate:

$$\Phi r_{t+1} = \text{ Proj. of } T(\Phi r_t) \text{ on Feature Subspace}$$

(with respect to some norm $\|\cdot\|$).

- **Convergence Result:** If T is a contraction with respect to a weighted Euclidean norm ($\|J\|^2 = J'DJ$, where D is positive definite, symmetric), then r_t converges to (the unique) r^* satisfying

$$r^* = \arg\min_r \left\| \Phi r - T(\Phi r^*) \right\|$$

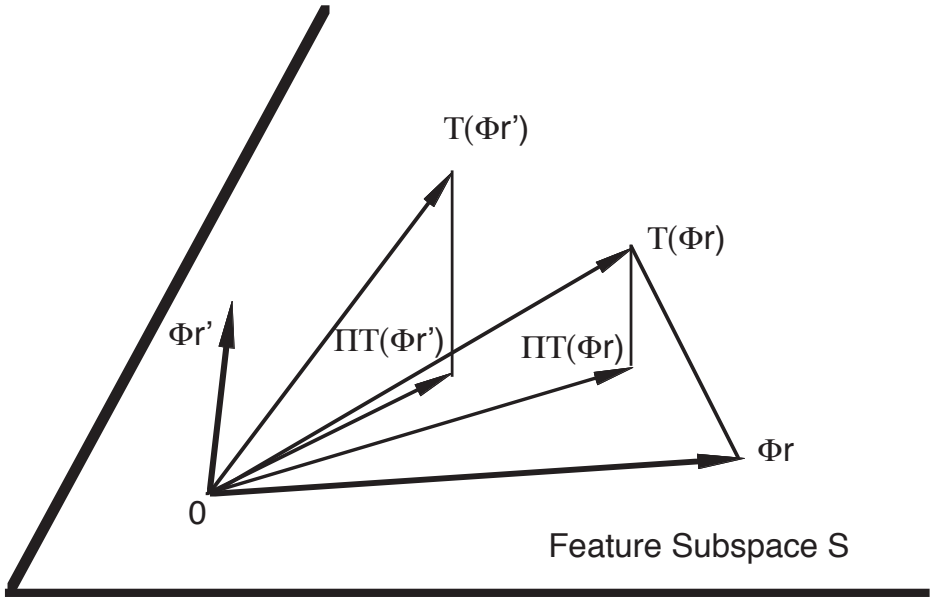
GEOMETRIC INTERPRETATION

- The approximate value iteration is

$$r_{t+1} = \Pi T(\Phi r_t)$$

where Π is projection on the subspace

$$S = \{\Phi r \mid r \in \Re^s\}$$



- **Convergence Proof Idea:** Since T is a contraction with respect to the norm of projection, and projection is nonexpansive, ΠT *is a contraction over* S (with respect to the same norm).

APPROXIMATE VALUE ITERATION BY SIMULATION

- The algorithm

$$r_{t+1} = \Pi T(\Phi r_t) = \arg\min_r \left\| \Phi r - T(\Phi r_t) \right\|^2$$

is deterministic, but is hard to implement for large number of states [we must compute $T(\Phi r_t)(i)$ for all i].

- We show, that with the right choice of norm, we can use simulation to implement the algorithm approximately, for a single policy.

- The simulation-based implementation solves a least squares problem at each iteration.

- Temporal differences (TD) are used to simplify the formulas of the algorithm.

USING A DIAGONALLY WEIGHTED EUCLIDEAN NORM

- Consider the approximate value iteration using a diagonally weighted Euclidean norm

$$r_{t+1} = \arg\min_r \sum_{i=1}^n w(i) \left(\phi(i)'r - \sum_{j=1}^n p_{ij} \Big(g(i,j) + \alpha \phi(j)' r_t \Big) \right)^2$$

where the $w(i)$ are some positive weights.

- This is a special case of

$$r_{t+1} = \Pi T(\Phi r_t) = \arg\min_r \big\| \Phi r - T(\Phi r_t) \big\|^2$$

corresponding to the norm defined by the $w(i)$.

- **Key fact:** T is a contraction with respect to the weighted Euclidean norm corresponding to the weights

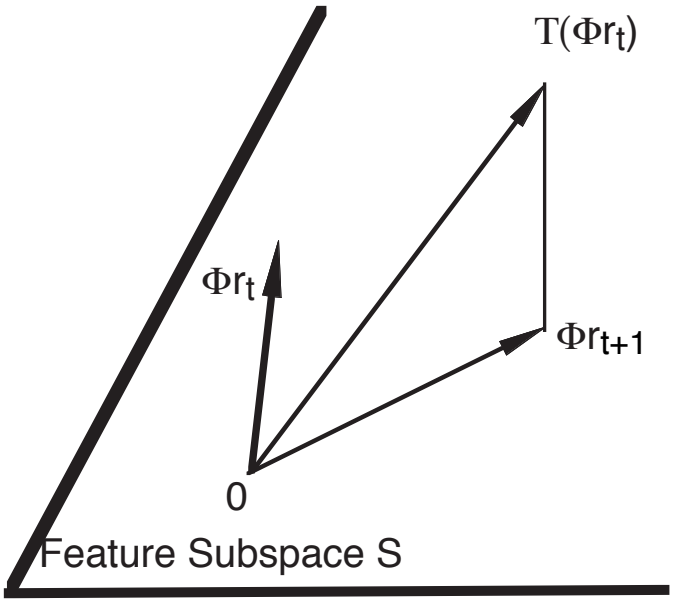
$$w(i) = \text{ the steady state probability of state } i$$

(Bertsekas and Tsitsiklis, NDP book, 1996).

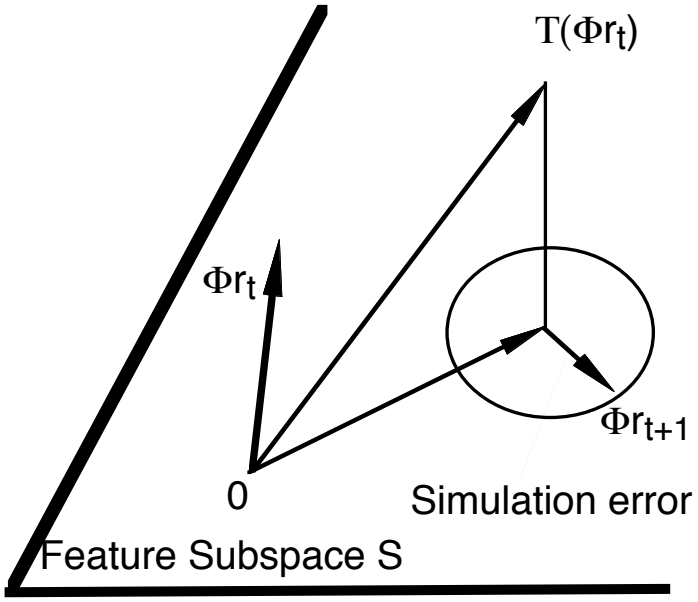
SIMULATION-BASED IMPLEMENTATION

- Generate an infinitely long trajectory $(i_0, i_1, \ldots)$ using a simulator, and iteratively update r by

$$r_{t+1} = \arg\min_r \sum_{m=0}^{t} \underbrace{\Big(\phi(i_m)'r - g(i_m, i_{m+1}) - \alpha\phi(i_{m+1})'r_t\Big)^2}_{\text{squared value iteration error at time } m}$$



Value Iteration with Linear Function Approximation



Simulation-Based Value Iteration with Linear Function Approximation

- The iteration yields the (deterministic) approximate value iterate [i.e., the projection of $T(\Phi r_t)$] plus stochastic simulation error.

FROM $\lambda = 0$ TO $\lambda > 0$ – M-STEP VALUE ITERATION

- For $M \geq 1$, consider the equation

$$J(i) = E\left[\alpha^M J(i_M) + \sum_{k=0}^{M-1} \alpha^k g(i_k, i_{k+1}) \,\middle|\, i_0 = i\right]$$

- This is the equation $J = T(J)$ for a modified Markov chain where each transition corresponds to M transitions of the original.

- This equation has the same solution as the ordinary (one-step) equation

$$J(i) = E\big[g(i,j) + \alpha J(j)\big]$$

- The corresponding value iteration method is

$$J_{t+1}(i) = E\left[\alpha^M J_t(i_M) + \sum_{k=0}^{M-1} \alpha^k g(i_k, i_{k+1}) \,\middle|\, i_0 = i\right]$$

and can be similarly approximated by simulation.

SIMULATION-BASED M-STEP VALUE ITERATION

- The corresponding simulation-based least-squares implementation is

$$r_{t+1} = \arg\min_r \sum_{m=0}^{t} \left(\phi(i_m)'r - \alpha^M \phi(i_{m+M})' r_t - \sum_{k=0}^{M-1} \alpha^k g(i_{m+k}, i_{m+k+1}) \right)^2$$

- Introduce the temporal differences,

$$d_t(i_k, i_{k+1}) = g(i_k, i_{k+1}) + \alpha\phi(i_{k+1})'r_t - \phi(i_k)'r_t,$$

to write this iteration as

$$r_{t+1} = \arg\min_r \sum_{m=0}^{t} \left(\phi(i_m)'r - \phi(i_m)'r_t - \sum_{k=m}^{m+M-1} \alpha^{k-m} d_t(i_k, i_{k+1}) \right)^2$$

USING RANDOM STEP VALUE ITERATION

- Consider a version of Bellman's equation where M is random and geometrically distributed with parameter λ, i.e.,

$$\text{Prob}(M = m) = (1-\lambda)\lambda^{m-1}, \qquad m = 1, 2, \ldots$$

- This equation is obtained by multiplying both sides of the M-step Eq. with $(1-\lambda)\lambda^{m-1}$, for each m, and adding over m:

$$J(i) = \sum_{m=1}^{\infty} (1-\lambda)\lambda^{m-1} E\left[\alpha^m J(i_m) + \sum_{k=0}^{m-1} \alpha^k g(i_k, i_{k+1}) \mid i_0 = i\right]$$

- The corresponding value iteration method is

$$J_{t+1}(i) = \sum_{m=1}^{\infty} (1-\lambda)\lambda^{m-1} E\left[\alpha^m J_t(i_m) + \sum_{k=0}^{m-1} \alpha^k g(i_k, i_{k+1}) \mid i_0 = i\right]$$

TEMPORAL DIFFERENCES IMPLEMENTATION

- We can write the random step value iteration as

$$J_{t+1}(i) = J_t(i) + \sum_{k=0}^{\infty} (\alpha\lambda)^k E\Big[g(i_k, i_{k+1}) + \alpha J_t(i_{k+1}) - J_t(i_k) \mid i_0 = i$$

- By using $\phi(i)'r_t$ to approximate J_t, we obtain the λ-least squares policy evaluation method, proposed by Bertsekas and Ioffe (1996)

$$r_{t+1} = \arg\min_r \sum_{m=0}^{t} \left(\phi(i_m)'r - \phi(i_m)'r_t - \sum_{k=m}^{t} (\alpha\lambda)^{k-m} d_t(i_k, i_{k+1}) \right)^2$$

- Role of TD: They simplify the formulas.

GENERIC PROPERTIES

- The limit r^* depends on λ, and Φr^* tends to be closer to J as $\lambda \approx 1$.

- But as $\lambda \approx 1$, the simulation noise is magnified, and convergence can be very slow.

- Using $\lambda < 1$ is essential for some "high noise" problems.

- Connection with TD(λ): We can view TD(λ) as a gradient-like iteration for minimizing the least-squares sum in the λ-least squares method.

CONVERGENCE ANALYSIS

- The method converges for all $\lambda \in [0,1]$ and with a stepsize equal to 1! (or any constant stepsize in $(0,1]$).

- This is the first TD method that does not require a diminishing $(1/k)$ type of stepsize.

- It bears to TD(λ) the same type of relation as the Kalman filter/Gauss-Newton algorithm bears to the gradient method.

- Converges dramatically faster than TD(λ), based on experimental and theoretical analysis (Yu and Bertsekas, 2006).

RATE OF CONVERGENCE ANALYSIS

- Konda (Ph.D. Thesis, MIT, 2002) has established the optimal convergence rate for TD methods.

- Konda has shown that λ-LSTD, a nonincremental method proposed by Bradtke-Barto (1996), Boyan (2002) attains the optimal rate.

- Yu and Bertsekas (2006) show that the iterates of λ-LSPE and λ-LSTD converge to each other faster than to the common limit.

- Hence λ-LSPE also attains the optimal rate.

- λ-LSPE has the advantage that it is an incremental method, and hence is more suitable for policy iteration contexts where there is limited simulation between policy updates (e.g., optimistic policy iteration).