New Sampling Schemes for Simulation-Based Approximate Dynamic Programming

Dimitri P. Bertsekas
joint work with
Huizhen Yu

Department of Electrical Engineering and Computer Science
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology

Outline

A Class of Generalized Bellman Equations

Ordinary Bellman equation for a policy μ of an n-state MDP

$$J = TJ$$

where

$$(TJ)(i) \stackrel{\text{def}}{=} \sum_{j=1}^{n} p_{ij}\big(\mu(i)\big)\big(g(i,\mu(i),j) + \alpha J(j)\big), \qquad i = 1,\ldots,n$$

$p_{ij}(u)$: transition probs, $g(i,u,j)$: cost per stage, α: discount factor

Generalized Bellman equation

$$J = T^{(w)}J$$

where w is a matrix of weights $w_{i\ell}$:

$$(T^{(w)}J)(i) \stackrel{\text{def}}{=} \sum_{\ell=1}^{\infty} w_{i\ell}(T^{\ell}J)(i), \qquad w_{i\ell} \geq 0, \ \sum_{\ell=1}^{\infty} w_{i\ell} = 1 \quad (\text{for each } i = 1,\ldots,n)$$

Both can be solved for J_μ, the cost vector of policy μ.

TD(λ) Special Cases

Classical TD(λ) mapping, $\lambda \in [0, 1)$

$$T^{(\lambda)}J = (1-\lambda)\sum_{\ell=1}^{\infty}\lambda^{\ell-1}T^{\ell}J, \qquad w_{i\ell} = (1-\lambda)\lambda^{\ell-1}$$

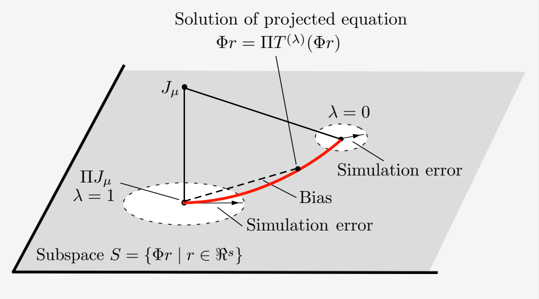
A generalization: State-dependent $\lambda_i \in [0, 1)$

$$(T^{(\lambda)}J)(i) = (1-\lambda_i)\sum_{\ell=1}^{\infty}\lambda_i^{\ell-1}(T^{\ell}J)(i), \qquad w_{i\ell} = (1-\lambda_i)\lambda_i^{\ell-1}$$

Generalized Bellman Eqs with Subspace Projection: $\Phi r = \Pi T^{(w)}(\Phi r)$

- Φ is an $n \times s$ matrix of features, defining subspace $S = \{\Phi r \mid r \in \Re^s\}$, $r \in \Re^s$ is a vector of weights.
- Π is projection onto S with respect to a weighted Euclidean semi-norm $\|J\|_\xi^2 = \sum_{i=1}^n \xi_i \big(J(i)\big)^2$, where $\xi = (\xi_1, \ldots, \xi_n)$, with $\xi_i \geq 0$.
- If $\|\cdot\|_\xi$ is a norm, this is Galerkin approximation specialized to DP.

Example: TD(λ) $\quad T^{(\lambda)}J = (1-\lambda)\sum_{\ell=1}^{\infty} \lambda^{\ell-1} T^\ell J, \qquad \lambda \in [0,1)$



Generalized Bellman Eqs with Aggregation

Aggregation case (r is the cost vector of an “aggregate" problem)

$r = DT^{(w)}(\Phi r)$, (low-dimensional) $\Phi r = \Phi DT^{(w)}(\Phi r)$, (high-dimensional)

where Φ and D are nonnegative matrices whose rows are prob. distributions.

Comparison with projection case

$$\Phi r = \Pi T^{(w)}(\Phi r)$$

Aggregation is a special case of projection if ΦD is a semi-norm projection.

First Benefit of the Generalization

$$\Phi r = \Pi T^{(w)}(\Phi r)$$

State-dependent weights $w_{i\ell}$

- Similar approximation properties as the TD(λ) mapping $T^{(\lambda)}$ (control the bias-variance tradeoff)
- New sampling schemes based on multiple short simulation trajectories (free form sampling)
- They control more flexibly the bias-variance tradeoff
- They naturally introduce exploration of the potential of other policies (in the context of policy iteration)

Second Benefit of the Generalization

Semi-norm projection - Can have $\xi_i = 0$

- More flexibility in simulation (some states need not be visited)
- Aggregation and projected equations become strongly connected if semi-norm projection is allowed
- Use of semi-norm allows (for the first time) multistep aggregation methods - analogs of TD(λ), LSTD(λ), LSPE(λ)

References

- H. Yu and D. P. Bertsekas, “Weighted Bellman Equations and their Applications in Approximate Dynamic Programming," Report LIDS-P-2876, MIT, 2012.
- D. P. Bertsekas, “λ-Policy Iteration: A Review and a New Implementation," Report LIDS-P-2874, MIT, 2011; in *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, by F. Lewis and D. Liu (eds.), IEEE Press, Computational Intelligence Series.
- D. P. Bertsekas, Dynamic Programming and Optimal Control, Vol. II, 4th Edition: Approximate Dynamic Programming, Athena Scientific, Belmont, MA, 2012.

Projected Value Iteration for Projected Equation $\Phi r = \Pi T^{(w)}(\Phi r)$

Exact form of projected value iteration

$$\Phi r_{k+1} = \Pi T^{(w)}(\Phi r_k)$$

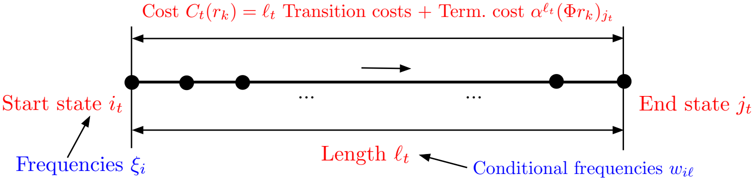
or

$$r_{k+1} = \arg\min_r \sum_{i=1}^{n} \xi_i \left(\phi(i)' r - \sum_{\ell=1}^{\infty} w_{i\ell} \big(T^{\ell}(\Phi r_k) \big)(i) \right)^2, \quad (\phi(i)'\text{: } i\text{th row of } \Phi)$$

We view the expression minimized as an expected value that can be simulated with Markov chain trajectories:

- ξ_i will be the "frequency" of i as start state of the trajectories
- $w_{i\ell}$ will be the "frequency" of trajectory length ℓ when i is the start state

Simulation-Based Implementation of Projected Value Iteration



Approximation using trajectories $t = 1, \ldots, m$

$$r_{k+1} = \arg\min_r \sum_{t=1}^{m} \big(\phi(i_t)'r - C_t(r_k)\big)^2$$

(i_t: start state, $C_t(r_k)$: sample cost)

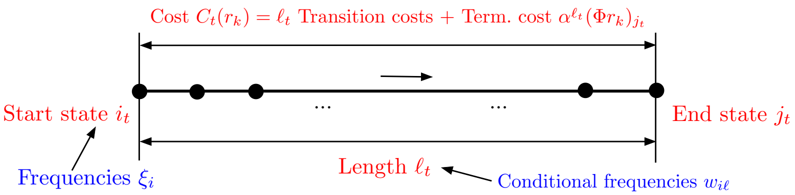
As freq. of start state $i \to \xi_i$, freq. of start-state/length $(i, \ell) \to \xi_i w_{i\ell}$

Opt. condition for simulation-based least squares

converges to

Opt. condition for exact least squares

Matrix Inversion Method (Extension of LSTD(λ))



Find $\hat{r}$ such that

$$\hat{r} = \arg\min_{r} \sum_{t=1}^{m} \big(\phi(i_t)' r - C_t(\hat{r})\big)^2$$

This is a linear system of equations (the equivalent optimality condition).

Example: Classical TD Sampling

$$T^{(\lambda)}J = (1-\lambda)\sum_{\ell=1}^{\infty}\lambda^{\ell-1}T^{\ell}J$$

- Generate one single infinitely long trajectory
- Segment it, and weigh the segments of length ℓ with geometric weights $w_{i\ell} = (1-\lambda)\lambda^{\ell-1}$
- Use the Markov chain invariant distribution as weight vector $\xi = (\xi_1, \ldots, \xi_n)$
- Requires modifications to deal with transient states and exploration (an off-policy scheme and modified TDs)

New Sampling Schemes

Geometric sampling

$$T^{(\lambda)}J = (1-\lambda)\sum_{\ell=1}^{\infty}\lambda^{\ell-1}T^{\ell}J$$

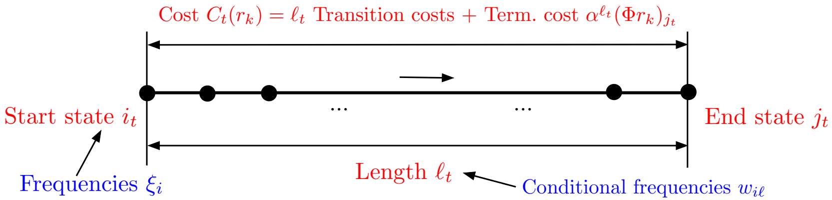
- Generate many short trajectories with random/geometrically distributed length (parameter λ, the same for all start states)
- Arbitrary restart distribution ξ. Provides implementation of LSPE(λ) and LSTD(λ) with exploration.

Free-form sampling

$$(T^{(w)}J)(i) \stackrel{\text{def}}{=} \sum_{\ell=1}^{\infty} w_{i\ell}(T^{\ell}J)(i), \qquad w_{i\ell} \geq 0,\ \sum_{\ell=1}^{\infty} w_{i\ell} = 1 \quad \text{(for each } i = 1,\ldots,n)$$

Anything goes as long as
freq. of start state $i \to \xi_i$, freq. of start-state/length $(i,\ell) \to \xi_i w_{i\ell}$

Free-Form Sampling



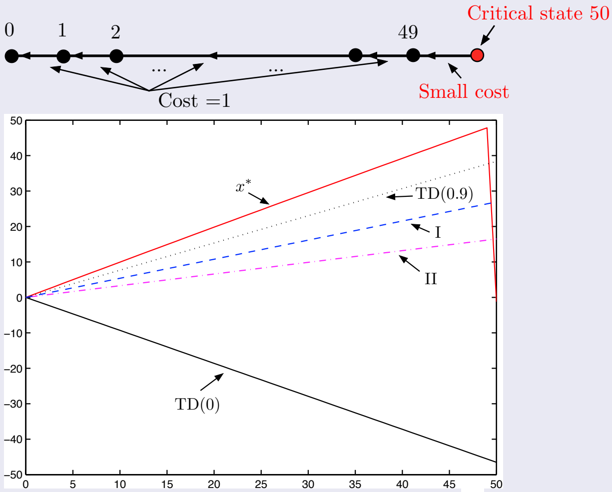
- Trajectories can be segmented into overlapping pieces, and even duplicated, to create extra shorter trajectories.
- Deals well with exploration.
- Lengths of trajectories can be dependent on the start state.
- Controls more flexibly the bias-variance tradeoff

Long segments $<->$ Large sample variance

- Can use large $w_{i\ell}$ for large ℓ selectively for some critical states i to reduce bias.
- Some weights may have “partially deterministic form" rather than be fully simulated.

Selective Bias-Variance Control: An Example

An example where TD(0) gives large bias and TD(1) large variance



One-stage costs are weighted disproportionally to future costs in TD(0).

Block TD(λ)-Type Algorithm: An Example

Use an upper bound m on the length of trajectories

- This makes sense if few samples are collected before changing policies (optimistic PI).
- We can use geometrically weighted coefficients (same $\lambda \in (0,1)$ for all i)

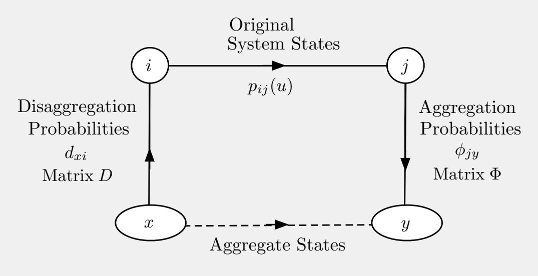
$$w_{i\ell} = (\text{normalization const}) \cdot \lambda^{\ell-1}, \qquad \ell = 1, \ldots, m$$

- The geometrically weighted coefficients can be state-dependent

$$w_{i\ell} = (\text{normalization const}) \cdot \lambda_i^{\ell-1}, \qquad \ell = 1, \ldots, m$$

  This allows more flexible control of the bias-variance tradeoff.
- Note the $w_{i\ell}$ are “partially deterministic" (less noise).
- Exploration is allowed through trajectory restarts to control the weights ξ_i.

Aggregation Framework



- Introduce s aggregate states, aggregation and disaggregation probabilities
- They define a s-dimensional aggregate Markov chain with single step Bellman equation

$$r = DT(\Phi r)$$

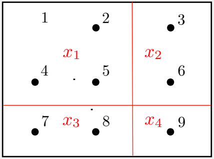
- Can obtain approximation Φr using the multistep versions

$$\Phi r = \Phi DT^{(\lambda)}(\Phi r) \qquad \text{or} \qquad \Phi r = \Phi DT^{(w)}(\Phi r)$$

which allow bias-variance tradeoff. If ΦD is a semi-norm projection the preceding methodology applies.

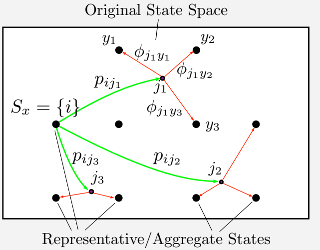
Two Common Types of Aggregation

- Hard aggregation: The aggregate states are disjoint subsets S_x of states with $\cup_x S_x = \{1, \ldots, n\}$, and $d_{xi} > 0$ only if $i \in S_x$, $\phi_{ix} = 1$ if $i \in S_x$.

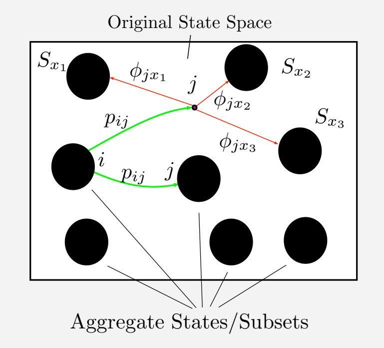


$$\Phi = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Aggregation with discretization grid of representative states: Each aggregate state is a single original system state $x \in \{1, \ldots, n\}$, and $d_{xx} = 1$.



A Generalization: Aggregation with Representative Features



- The aggregate states are disjoint subsets S_x of states
- Common case: S_x is a group of states with "similar features"
- Hard aggregation is a special case: $\cup_x S_x = \{1, \ldots, n\}$
- Aggregation with representative states is a special case: S_x consists of just one state

Connection with Semi-Norm Projection

- Assume that the approximation is piecewise constant with interpolation: constant within the aggregate states, interpolated for the other states, i.e., the disaggregation and aggregation probs satisfy

$$\phi_{ix} = 1 \ \ \forall \ i \in S_x, \qquad d_{xi} > 0 \ \text{ iff } \ i \in S_x$$

  Then ΦD is a semi-norm projection with

$$\xi_i = d_{xi}/s, \qquad \forall \ i \in S_x$$

- The use of a semi-norm is critical since $\xi_i = 0$ for $i \notin \cup_x S_x$ (except in the case of hard aggregation where $\xi_i > 0$ for all i).

Multistep Aggregation

- Multistep aggregation analogs of TD(λ), LSPE(λ), and LSTD(λ) are well-defined.
- Multistep aggregation with free-form sampling is well-defined.
  - Generate many short trajectories with the original system.
  - The start state of each trajectory must be in $\cup_x S_x$.
- A lot of flexibility for exploration.
- Flexible control of bias-variance tradeoff (use longer trajectories for “critical" start states).
- The multistep equation $\Phi r = \Phi D T^{(w)}(\Phi r)$ is a sup-norm contraction if T is.

Concluding Remarks

- Presented a class of generalized weighted Bellman equations.
- They allow state-dependent weights.
- They allow the use of a variety of sampling methods.
  - Flexible treatment of the bias-variance tradeoff.
- They allow semi-norm projection.
  - Connection between projected equations and aggregation equations.
- Also allows multistep aggregation methods of the TD(λ) type (but more general).
- The methodology extends to the much broader field of Galerkin approximation.