Some New Directions in Dynamic Programming with Cost Function Approximation

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

IEEE Symposium on ADPRL
December 2014

Outline

1. Three Interrelated Research Directions
   - Seminorm Projections (Unifying Projected Equation and Aggregation Approaches)
   - Generalized Bellman Equations (Multistep with State-Dependent Weights)
   - Free Form Sampling (A Flexible Alternative to Single Long Trajectory Simulation)
2. Aggregation and Seminorm Projected Equations
3. Simulation-Based Solution
   - Iterative and Matrix Inversion Methods
   - Free-Form Sampling

Bellman Equations and their Fixed Points

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

$$J = T_\mu J$$

where

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

$p_{ij}(\mu(i))$: transition probs, $g(i,\mu(i),j)$: cost per stage for μ

Bellman equation for the optimal cost function of an n-state MDP

$$J = TJ$$

where

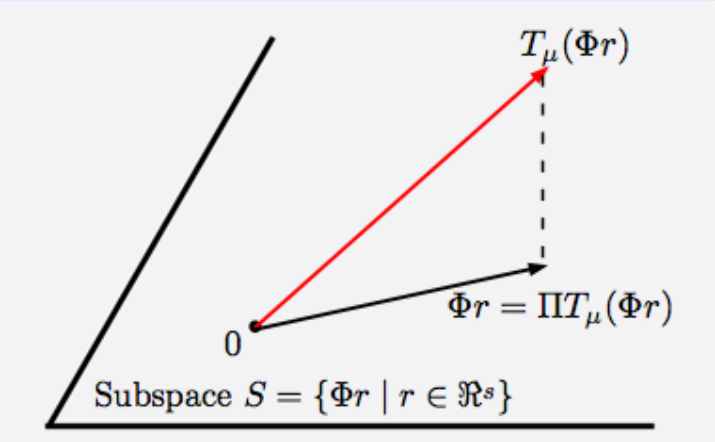
$$(TJ)(i) \stackrel{\text{def}}{=} \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u)\big(g(i,u,j) + \alpha J(j)\big), \qquad i = 1, \ldots, n$$

$p_{ij}(u)$: transition probs, $g(i,u,j)$: cost per stage for a control u

Subspace Approximation $J \approx \Phi r$ (Using a Matrix of Basis Functions Φ)

Methods with subspace approximation

- Projected equation (Galerkin) approach $\Phi r = \Pi T_\mu(\Phi r)$ (Π is projection with respect to some weighted Euclidean norm)
- Aggregation approach $\Phi r = \Phi D T_\mu(\Phi r)$ (Φ and D are matrices whose rows are probability distributions)
- Bellman error method ($\Phi r = \Pi \hat{T}_\mu(\Phi r)$, for a modified mapping $\hat{T}_\mu$ that has the same fixed points as T_μ)



First direction of research aims to connect all these

All of these can be written as $\Phi r = \Pi T_\mu(\Phi r)$, where Π is a seminorm weighted Euclidean projection

Another Direction of Research: Generalized Bellman Equations

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

$$J = T_\mu J$$

Generalized Bellman equation

$$J = T_\mu^{(w)} J$$

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

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

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

Two differences of generalized vs ordinary Bellman equations

- Multistep mappings (an old idea, e.g., TD(λ))
- State dependent weights (a new idea)

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 $\lambda_i \in [0, 1)$

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

Why state dependent weights?

- They may allow exploitation of prior knowledge for better approximation (emphasize important states)
- They may facilitate simulation (for special cases such as aggregation)

A Third Direction for Research: Flexible/Free-Form Simulation

Classical TD Sampling

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

- Simulate one single infinitely long trajectory, and move the starting state to generate multiple (infinitely long) trajectories
- This is well-matched to the structure of TD
- Does not work well in the aggregation context, where there are both regular and aggregate transitions (powers $T^{\ell}J$ involve ℓ regular transitions but no aggregate transitions)
- TD sampling matches well with regular transitions but not with aggregate transitions

Free-form sampling

- Generates many short trajectories (length ℓ $<->$ term $T^{\ell}J$)
- Arbitrary restart distribution
- Connects well with state-dependent weights (and allows restarting at an aggregate state in the case of aggregation)

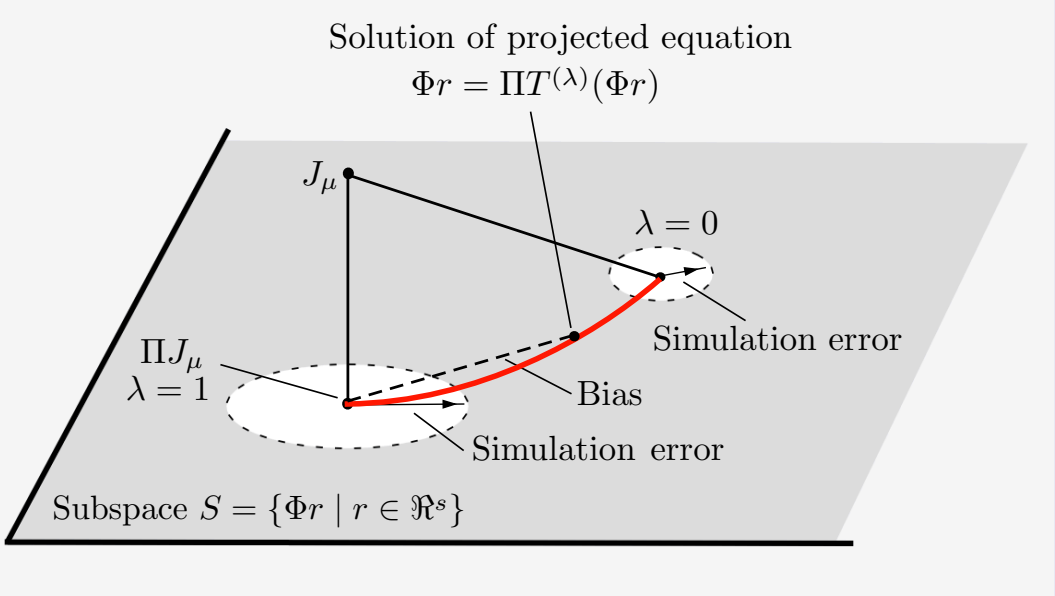
References

- D. P. Bertsekas, "λ-Policy Iteration: A Review and a New Implementation," in *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, by F. Lewis and D. Liu (eds.), IEEE Press, Computational Intelligence Series, 2012 (simulation with short trajectories and restart, as a means to control exploration).
- H. Yu and D. P. Bertsekas, "Weighted Bellman Equations and their Applications in Approximate Dynamic Programming," Report LIDS-P-2876, MIT, 2012 (weighted Bellman equations and seminorm projections).
- D. P. Bertsekas, Dynamic Programming and Optimal Control, Vol. II, 4th Edition: Approximate Dynamic Programming, Athena Scientific, Belmont, MA, 2012 (a general reference where all the ideas are mentioned with limited analysis).

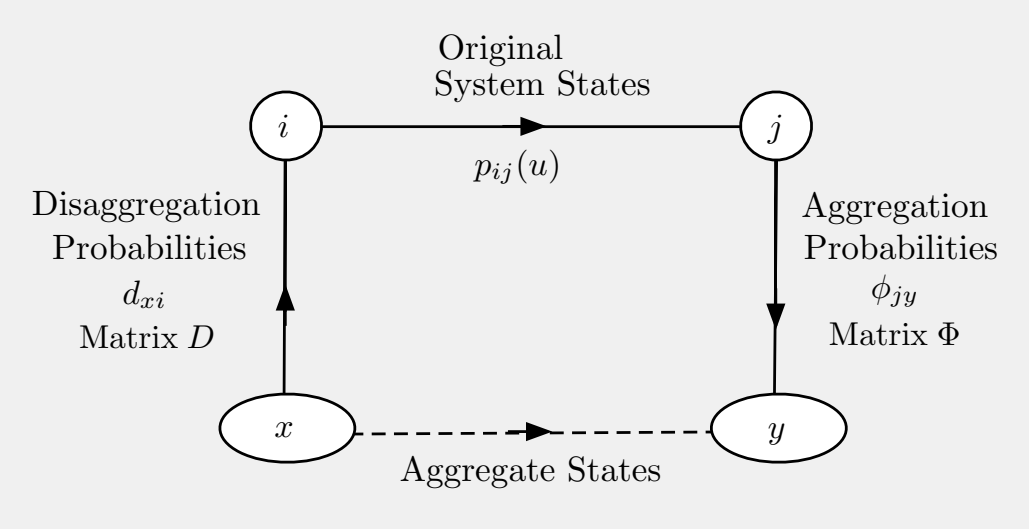
Generalized Bellman Eqs with Seminorm Projection: $\Phi r = \Pi T^{(w)}(\Phi r)$

- 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 seminorm $\|J\|_\xi^2 = \sum_{i=1}^n \xi_i \big(J(i)\big)^2$, where $\xi = (\xi_1, \ldots, \xi_n)$, with $\xi_i \geq 0$.
- Bias-variance tradeoff applies to both norm and seminorm cases.

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



Aggregation Framework



- Introduce s aggregate states, aggregation and disaggregation probs
- A composite system with both regular and aggregate states
- Two single step Bellman equations

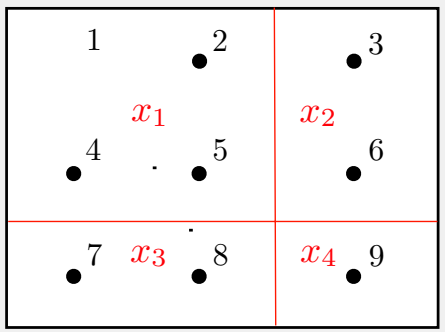
$$r = DT(\Phi r), \qquad \Phi r = \Phi DT(\Phi r)$$

  r is the cost vector of the aggregate states, Φr the cost vector of the regular states
- Natural multistep versions for bias-variance tradeoff:

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

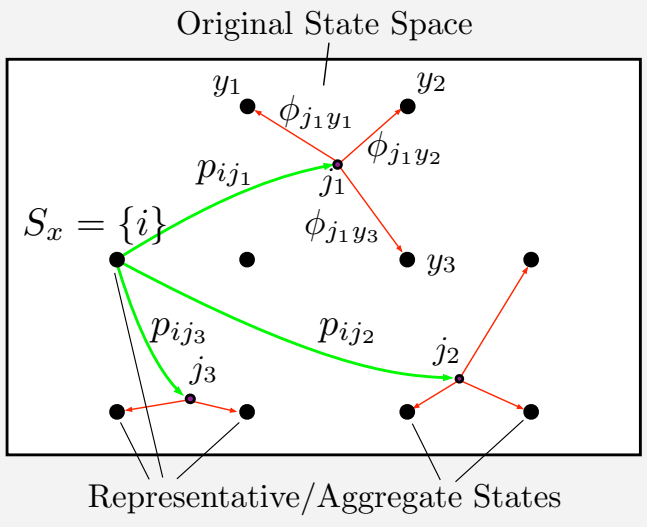
Two Common Types of Aggregation

- 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 $i \in S_x$.

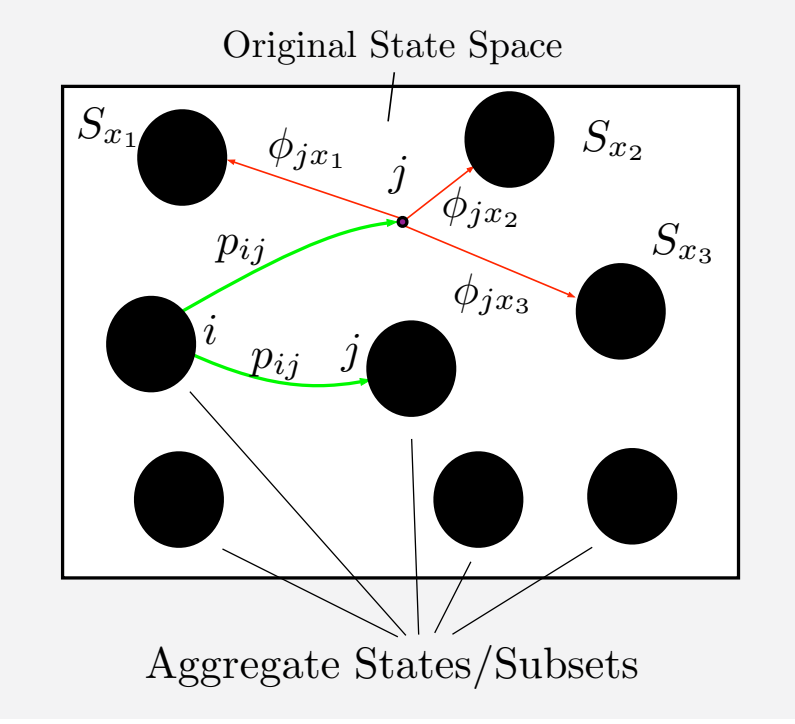


$$\Phi = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- 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 “similar" 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 Seminorm Projection

Consider the aggregation equations

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

Compare them with projected equation case $\Phi r = \Pi T^{(w)}(\Phi r)$

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

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

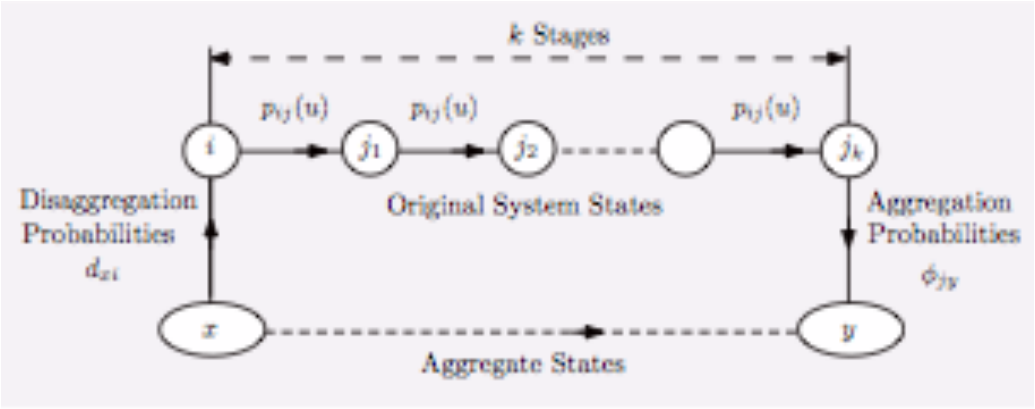
Then ΦD is a seminorm projection with

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

This is true for the preceding aggregation schemes. Moreover, the multistep equation $\Phi r = \Phi DT^{(w)}(\Phi r)$ is a sup-norm contraction if T is.

Sampling for Aggregation

- The classic form of TD sampling does not work for multistep aggregation.
- Reason: In aggregation we need to simulate multistep cost samples involving both regular and aggregate states. This cannot be easily done with classical TD sampling.
- So we introduce a more general (free-form) sampling.
- Generate many short trajectories.



- In aggregation, the start and end states of each trajectory must be an aggregate state.
- A side benefit: A lot of flexibility for “exploration".

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

Exact form of projected value iteration

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

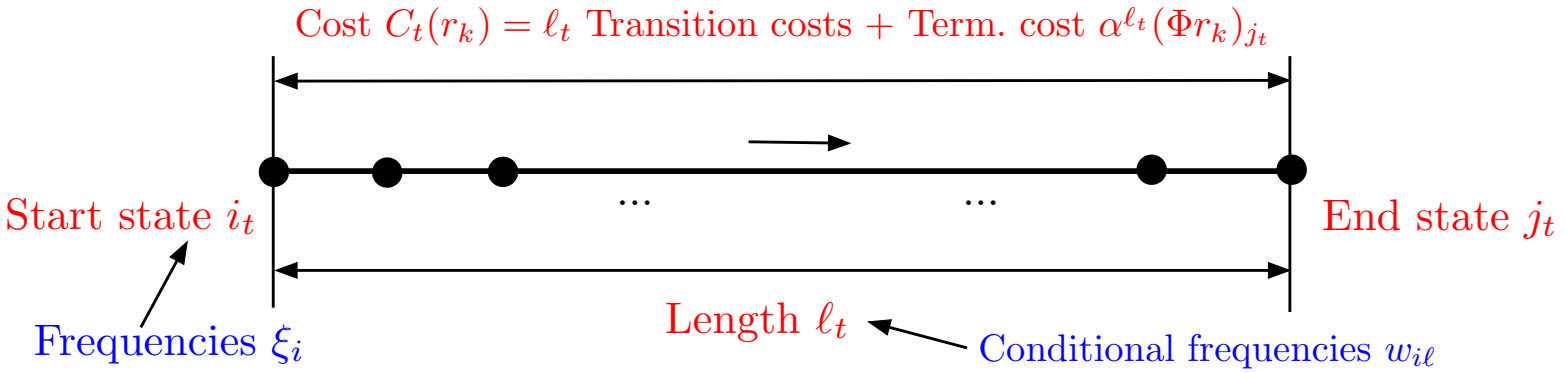
or

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

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 $t = 1, \ldots, m$

$$r_{k+1} = \arg\min_r \sum_{t=1}^{m} \left(\phi(i_t)'r - C_t(r_k)\right)^2 \quad (i_t\text{: start state, } C_t(r_k)\text{: sample cost})$$

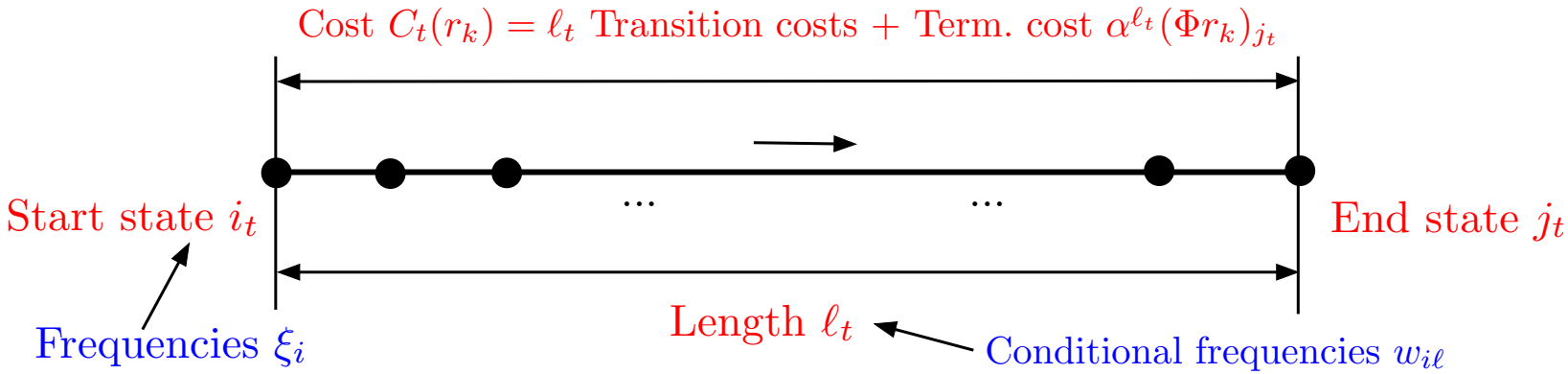
Since freq. of start state $i \to \xi_i$, freq. of start-state/length $(i, \ell) \to \xi_i w_{i\ell}$

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

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

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

Concluding Remarks

- Extension of cost function approximation methodology in DP via three interlocking ideas:
  - Seminorm projections.
  - Generalized weighted Bellman equations.
  - Free-form simulation.
- The approximation framework is general enough to include both multistep projected equations and aggregation (and other methods).
- Some of the highlights:
  - Connection between projected equations and aggregation equations.
  - Multistep aggregation methods of the TD(λ) type.
  - Use of a variety of sampling methods.
  - Flexible treatment of the bias-variance tradeoff.
- The methodology extends to the much broader field of Galerkin approximation for solving general linear equations.

Thank you!