Feature Selection and Basis Function Adaptation in Approximate Dynamic Programming

Dimitri P. Bertsekas

Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology

SSCI Workshop, Nashville 2009

DP Context

- Markovian Decision Problems (MDP)
- n states, transition probabilities depending on control
- Policy iteration method; we focus on single policy evaluation

- Bellman's equation:

$$x = Ax + b$$

where
  - b: cost vector
  - A has transition structure, e.g.

    $A = \alpha P$ for discounted problems; α: discount factor
    $A = P$ for average cost problems

Approximate Policy Evaluation

- Approximation within subspace $S = \{\Phi r \mid r \in \Re^s\}$

  $x \approx \Phi r$, Φ is a matrix with basis functions/features as columns

- Projected Bellman equation: $x = \Pi T(x)$ or

$$\Phi r = \Pi(A\Phi r + b)$$

- Long algorithmic history, starting with TD(λ) (Sutton, 1988)
- Least squares methods (LSTD, LSPE) are currently more popular

Focus

- Question: How do we select the matrix Φ?
- Subject of great importance, yet largely problem-specific, little understood at present.
- We will sample a few issues of interest, and describe some recent work.

References

- H. Yu and D. P. Bertsekas, “Basis Function Adaptation Methods for Cost Approximation in MDP," IEEE SSCI Conference Proceedings, 2009; following the paper by I. Menache, S. Manor, and N. Shimkin, Annals of OR, 2005.
- D. P. Bertsekas and H. Yu, “Projected Equation Methods for Approximate Solution of Large Linear Systems," Journal of Computational and Applied Mathematics, 2008.
- D. P. Bertsekas, “Projected Equations, Variational Inequalities, and Temporal Difference Methods," LIDS Report, MIT, 2009.

Overview of Questions Addressed

- Basis function tuning by gradient descent
  - Parametrize approximation subspace/basis functions by vector θ
    $$S_\theta = \{\Phi(\theta)r \mid r \in \Re^s\}$$
  - Compute the corresponding solution $x(\theta)$ of the projected equation $x = \Pi_\theta T(x)$, where Π_θ is projection on S_θ.
  - Optimize some cost function $F\big(x(\theta)\big)$ over θ (e.g., F is the Bellman equation error).
- Automatic basis function generation
  - Use a Krylov subspace basis for the equation $x = Ax + b$
    $$\Phi = [b \; Ab \; A^2b \; \cdots \; A^{s-1}b]$$
  - Problem is that for high-dimensional problems $A^k b$ cannot be computed
  - We use instead simulation-based samples of $A^k b$ (noisy features)
- Feature scaling
  - For a fixed subspace S, we consider alternative representations
    $$S = \{\Phi r \mid r \in \Re^s\} = \{\Psi v \mid v \in \Re^{\bar{s}}\}$$
    where $\Phi = \Psi B$.
  - Question: How are the popular algorithms for solving projected equations affected by such feature scaling?

Outline

Basis Function Tuning

- Parametrize approximation subspace/basis functions by vector θ

$$S_\theta = \left\{\Phi(\theta) r \mid r \in \Re^s\right\}$$

- Compute the corresponding solution $x(\theta)$ of the projected equation $x = \Pi_\theta T(x)$, where Π_θ is projection on S_θ, with respect to a weighted Euclidean norm (independent of θ).
- Optimize over θ the Bellman equation error:

$$F\big(x(\theta)\big) = \|x(\theta) - Ax(\theta) - b\|^2$$

- Key idea: Compute approximately gradient $\nabla F\big(x(\theta)\big)$ by simulation and low-order computation, and use a gradient-based method.
- For this we need an expression for the partial derivatives $\frac{\partial x(\theta)}{\partial \theta_j}$.

Computation of $\frac{\partial x(\theta)}{\partial \theta_j}$

- We differentiate the projected equation separately with respect to each component θ_j of θ

$$\frac{\partial x}{\partial \theta_j}(\theta) = \frac{\partial \Pi}{\partial \theta_j}(\theta) T(x(\theta)) + \Pi(\theta) A \frac{\partial x}{\partial \theta_j}(\theta)$$

- This can be simplified to

$$\frac{\partial x}{\partial \theta_j}(\theta) = \frac{\partial \Phi}{\partial \theta_j}(\theta) r(\theta) + \Phi(\theta) \frac{\partial r}{\partial \theta_j}(\theta)$$

  with $\Phi(\theta) r(\theta) = x(\theta)$.

- The second component $\Phi(\theta)\frac{\partial r}{\partial \theta_j}(\theta)$ of the derivative is the solution of the "projected" equation

$$y = \Pi(\theta)\big(Ay + q_j(x)\big)$$

  where the vector $q_j(x) \in S_\theta$ is given by

$$q_j(x) = \frac{\partial \Pi}{\partial \theta_j}(\theta)\big(Ax(\theta) + b\big) + \big(\Pi(\theta)A - I\big)\frac{\partial \Phi}{\partial \theta_j}(\theta) r(\theta)$$

Comments

- Each partial derivative $\frac{\partial x(\theta)}{\partial \theta_j}$ requires the solution of a separate projected equation.
- Each projected equation can be solved by simulation and low-dimensional calculations.
- We could replace T with a multistep Bellman operator $T^{(\lambda)}$, $\lambda \in (0,1)$.
- We may use a cost function different than the Bellman error criterion, e.g.,

$$F(x(\theta)) = \frac{1}{2} \sum_{i \in \mathcal{I}} \big(J_i - x_i(\theta)\big)^2,$$

  where $\mathcal{I}$ is a certain small subset of states, and J_i, $i \in \mathcal{I}$, are the costs of the policy at these states calculated directly by simulation.
- Potential difficulties: Slow convergence, nonconvex cost/local minima.

Automatic Generation of Powers of A as Basis Functions

- Use Φ whose ith row is

$$\phi(i)' = \big(b(i)\ (Ab)(i)\ \cdots\ (A^{s-1}b)(i)\big)$$

  (the Krylov subspace basis).
- Thus, we use as features finite horizon expected costs, i.e., $(A^{k-1}b)(i)$ is the k-stage vector starting at state i.
- Motivation: If A is a contraction, the fixed point of T has an expansion of the form

$$x^* = \sum_{k=0}^{\infty} A^k b$$

- While $(A^k b)(i)$ is hard to generate, it can be approximated by sampling (in effect we use noisy features).
- Features $A^k b$ may be supplemented with other “noiseless" features.

Implementation Within TD methods

- Simulate the Markov chain to obtain a sequence of states $\{i_0, i_1, \ldots\}$, as is usual in TD.
- At each generated state i_k, we also generate two additional mutually "independent" sequences

$$\big\{(i_k, \hat{i}_{k,1}), (\hat{i}_{k,1}, \hat{i}_{k,2}), \ldots, (\hat{i}_{k,s-1}, \hat{i}_{k,s})\big\}, \qquad \big\{(i_k, \tilde{i}_{k,1}), (\tilde{i}_{k,1}, \tilde{i}_{k,2}), \ldots, (\tilde{i}_{k,s}, \tilde{i}_{k,s+1})\big\}$$

  of lengths s and $s+1$, respectively, according to the transition probabilities p_{ij}, which are also "independent" of the sequence $\{i_0, i_1, \ldots\}$.
- The two extra sequences give single sample approximations of the basis function components $b(i_k), (Ab)(i_k), \ldots, (A^s b)(i_k)$.
- The single samples are used in the TD/LSTD/LSPE formulas as if they were exact averages.
- Convergence properties are maintained, but noise in the algorithms is increased.

Alternative Representations of Approximation Subspace

- For a fixed subspace S, we consider alternative representations

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

  where $\Phi = \Psi B$ (B is a matrix whose range contains the range of Ψ').
- Consider the high-dimensional sequences Φr_k and Ψv_k generated by TD methods.
- The high-dimensional sequences generated by LSTD and LSPE are scale-free (do not depend on the representation of S).
- The high-dimensional sequences generated by LSPE with direction scaling are asymptotically scale-free.
- The convergence of TD methods does not depend on Φ having full rank!

Conclusions

- Feature selection is an important and multi-faceted problem.
- Many researchers have contributed to it, but many challenges remain.
- We discussed a few selected approaches and issues.
- Much remains to be done ...