ECE 555 Control of Stochastic Systems Fall 2005

Handout: ***More on reinforcement learning: Value function approximation***

In this handout we introduce methods to approximate the value function for a given policy for application reinforcement learning algorithms for Markov decision processes. The reader is again referred to [2, 8] for a general background. More detailed treatments of temporal difference (TD) learning and value function approximation can be found in [9, 7, 6, 1].

Throughout this handout we let $\boldsymbol{X}$ denote a Markov chain without control on a state space X with transition matrix P, and unique invariant distribution π. A cost function $c\colon \mathsf{X} \to \mathbb{R}$ is given, and our goal is to estimate the solution to the DP equation,

$$Ph^* - (1+\gamma)h^* + c = 0. \tag{1}$$

We restrict to the finite state space case with $\mathsf{X} = \{1, \dots, s\}$ to simplify notation.

The issues addressed in this handout are summarized in the following remark from [5] in discussing practical issues in the implementation of MDP methods:

> A large state space presents two major issues. The most obvious one is the storage problem, as it becomes impractical to store the value function (or optimal action) explicitly for each state. The other is the generalization problem, assuming that limited experience does not provide sufficient data for each and every state.

These issues are each addressed by constructing an approximate solution to (1) over a parameterized set of functions.

Linear approximations Suppose that $\{\psi_i : 1 \le i \le q\}$ are functions on X. We seek a best fit among a set of parameterized approximations,

$$h^r(x) := r^{\mathrm{T}}\psi(x) = \sum_{i=1}^{q} r_i\psi_i(x), \qquad x \in \mathsf{X}.$$

To choose r we first define a particular metric to describe the distance between h^r and h^*. There are many ways to do this - an L_2 setting leads to an elegant solution. In this finite state space setting we view h^r and h^* as column vectors in $\mathbb{R}^s$. For a given $s \times s$ matrix M we define the L_2 error $\|h^r - h^*\|_M^2 = (h^r - h^*)^{\mathrm{T}} M (h^r - h^*)$. We will focus on the special case $M = \mathrm{diag}(\pi(1), \dots, \pi(s))$ so that the L_2 error can be expressed,

$$\|h^r - h^*\|_M^2 = \mathsf{E}_\pi[(h^r(X(k)) - h^*(X(k)))^2] = \mathsf{E}_\pi[(r^{\mathrm{T}}\psi(X(k)) - h^*(X(k)))^2].$$

The derivative with respect to r has the probabilistic interpretation,

$$\nabla_r \|h^r - h^*\|_M^2 = 2\mathsf{E}_\pi[(r^{\mathrm{T}}\psi(X(k)) - h^*(X(k)))\psi(X(k))], \tag{2}$$

and setting this equal to zero gives the optimal value,

$$r^* = A^{-1}b, \qquad \text{where } A = \mathsf{E}_\pi[\psi(X)\psi(X)^{\mathrm{T}}], \quad b = \mathsf{E}_\pi[h^*(X)\psi(X)].$$

We assume henceforth that $A > 0$.

The steepest descent algorithm to compute r^* is given by,

$$\tfrac{d}{dr} r(t) = -a\nabla_r \|h^r - h^*\|^2_M = -a[Ar + b], \qquad t \ge 0, \tag{3}$$

where $a > 0$ is a gain. Although this leads to a natural stochastic approximation algorithm, the function h^* appearing in the definition of b is not known. Given the representation $h^* = R_\gamma c := [(1+\gamma)I - P]^{-1}c$, we could resort to the pair of O.D.E.s,

$$\begin{aligned} \tfrac{d}{dt} r &= -Ar + \pi(h\psi) \\ \tfrac{d}{dt} h &= Ph - (1+\gamma)h + c \end{aligned} \tag{4}$$

This is exponentially asymptotically stable when $M > 0$. Since $M = \mathrm{diag}(\pi)$, this amounts to irreducibility of $\boldsymbol{X}$. The following S.A. recursion follows naturally

$$\begin{aligned} r(k+1) - r(k) &= a_k \sum_{i=1}^{s} \mathbb{I}\{X(k) = i\}[h(i;k) - \psi^{\mathrm{T}}(i) r(k)]\psi(i) \\ h(i;k+1) - h(i;k) &= a_k \mathbb{I}\{X(k) = i\}[h(X(k+1);k) - (1+\gamma)h(i;k) + c(i)], \qquad i \in \mathsf{X}, \end{aligned} \tag{5}$$

where $\{a_k\}$ is a vanishing gain sequence. The corresponding ODE is almost (4) except that the h equation is modified,

$$\tfrac{d}{dt} h(i;t) = \pi(i)[Ph\,(t;i) - (1+\gamma)h(t;i) + c(i)], \qquad i \in \mathsf{X}.$$

Since this evolves autonomously and is linear, analysis of the two coupled ODEs is straightforward.

The algorithm (5) may remain too complex for application in large problems. Observe that it is necessary to maintain estimates of $h^*(i)$ for each $i \in \mathsf{X}$, which means that the memory requirements are linear in the size of X. A simple remedy can be found through a closer look at the derivative equation (2).

L_2 theory The right hand side of (2) can be written, $\frac{d}{dr}\|h^r - h^*\|^2_M = 2\pi(fg)$, with $f = h^r - h^*$ and $g = \psi$. The resolvent $R_\gamma c$ will be transformed in the representation $h^* = R_\gamma c$ using some duality theory.

Consider the Hilbert space $L_2(\pi)$ consisting of real-valued functions on X whose second moment under π is finite. This simply means the function is finite-valued in the finite state space case. For $f, g \in L_2(\pi)$ we define the inner product,

$$\langle f, g \rangle = \pi(fg).$$

The *adjoint* $\widetilde{R}_\gamma$ of the resolvent is characterized by the defining set of equations,

$$\langle R_\gamma f, g \rangle = \langle f, \widetilde{R}_\gamma g \rangle, \qquad f, g \in L_2(\pi).$$

To construct the adjoint we obtain a sample path representation for $\langle R_\gamma f, g \rangle$. Let $\boldsymbol{X}$ denote a stationary version of the Markov chain on the two sided interval. We have,

$$\langle R_\gamma f, g \rangle = \mathsf{E}\Big[\Big(\sum_{t=0}^{\infty} (1+\gamma)^{-t-1} P^t f\,(X(0))\Big) g(X(0))\Big]$$

We have by the smoothing property of the conditional expectation,

$$\mathsf{E}[P^t f\,(X(0)) g(X(0))] = \mathsf{E}\big[\mathsf{E}[f(X(t)) \mid X(0)] g(X(0))\big] = \mathsf{E}[f(X(t)) g(X(0))]$$

and then applying stationarity of $\boldsymbol{X}$ and the smoothing property once more,

$$\mathsf{E}[P^t f\,(X(0))g(X(0))] = \mathsf{E}[f(X(0))g(X(-t))] = \sum \pi(x)f(x)\mathsf{E}[g(X(-t)) \mid X(0)=x].$$

Consequently,

$$\langle R_\gamma f, g\rangle = \sum_{t=0}^{\infty}(1+\gamma)^{-t-1}\mathsf{E}[f(X(0))g(X(-t))] = \langle f, \widetilde{R}_\gamma g\rangle,$$

where the adjoint is expressed,

$$\widetilde{R}_\gamma g\,(x) = \sum_{t=0}^{\infty}(1+\gamma)^{-t-1}\mathsf{E}[g(X(-t)) \mid X(0)=x]. \tag{6}$$

Applying the adjoint equation to the definition of b given below (2) gives,

$$b = \mathsf{E}_\pi[h^*(X(k))\psi(X(k))] = \mathsf{E}_\pi[R_\gamma c\,(X(k))\psi(X(k))] = \mathsf{E}_\pi[c(X(k))\widetilde{R}_\gamma\,\psi(X(k))]$$

Based on (6) we obtain,

$$b = \sum_{t=0}^{\infty}(1+\gamma)^{-t-1}\mathsf{E}_\pi[c(X(k))\psi(X(k-t))]. \tag{7}$$

This final representation (7) is the basis of TD learning.

Temporal difference learning Returning to (2) we have,

$$\nabla_r\|h^r - h^*\|_M^2 = 2\langle h^r - h^*, \psi\rangle = 2\langle h^r - R_\gamma c, \psi\rangle$$

and writing $h^r - R_\gamma c = R_\gamma[(1+\gamma)h^r - Ph^r - c]$ we obtain from the adjoint equation,

$$\nabla_r\|h^r - h^*\|_M^2 = 2\big\langle (1+\gamma)h^r - Ph^r - c, \widetilde{R}_\gamma\psi\big\rangle \tag{8}$$

Written as an expectation we obtain

$$\nabla_r\|h^r - h^*\|_M^2 = 2\mathsf{E}\big[[(1+\gamma)h^r(X(k)) - h^r(X(k+1)) - c(X(k))][\widetilde{R}_\gamma\psi\,(X(k))]\big] \tag{9}$$

We now have sufficient motivation to construct the TD learning algorithm based on the O.D.E. (3). The algorithm constructs recursively a sequence of estimates $\{r(k)\}$ based on the following,

(i) The *temporal differences* in TD learning are defined by,

$$d(k) := -[(1+\gamma)h^{r(k)}(X(k)) - h^{r(k)}(X(k+1)) - c(X(k))]\,. \tag{10}$$

(ii) *Eligibility vectors* are the sequence of q-dimensional vectors,

$$z(k) = \sum_{t=0}^{k}(1+\gamma)^{-t-1}\psi(X(k-t))\,, \qquad k \ge 1,$$

expressed recursively via,

$$z(k+1) = (1+\gamma)^{-1}[z(k) + \psi(X(k+1))], \qquad k \ge 0,\ z(0) = 0.$$

Since $\boldsymbol{X}$ is ergodic we have for any $g\colon \mathsf{X} \to \mathbb{R}$,

$$\lim_{k\to\infty} \mathsf{E}[g(X(k))z(k)] = \langle g, \widetilde{R}_\gamma \psi\rangle.$$

Based on (9), for large k we obtain the approximation,

$$\mathsf{E}[d(k)z(k+1)] \approx -\tfrac{1}{2}\nabla_r \|h^r - h^*\|_M^2, \qquad r = r(k).$$

The TD algorithm is the stochastic approximation algorithm associated with the O.D.E. (3),

$$r(k+1) - r(k) = a_k d(k) z(k+1), \qquad k \ge 0. \tag{11}$$

The O.D.E. (3) is linear and exponentially asymptotically stable under the assumption that $A = \mathsf{E}_\pi[\psi(X)\psi(X)^{\mathrm{T}}] > 0$. Based on this fact, one can show that the sequence of estimates $\{r(k)\}$ obtained from the TD algorithm (11) is convergent for the vanishing step-size algorithm.

Extensions *Where to begin?* There is the issue of constructing the basis functions $\{\psi_i\}$ [5]. One can also extend these methods to construct an approximation based on a family of non-linearly parameterized functions $\{h^r\}$ [2, 8]. Below are a few extensions in the case of linear approximations.

- The most common extension found in the literature is to redefine the definition of $\{z(k)\}$. Fix any $\lambda \in [0,1]$ and consider the new definition,

$$z(k+1) = (1+\gamma)^{-1}[\lambda z(k) + \psi(X(k+1))], \qquad z(0) = 0.$$

The resulting algorithm (11) is called TD(λ), where the definition of the temporal differences remain unchanged. In particular, TD(0) takes the form,

$$r(k+1) - r(k) = a_k d(k) \psi(X(k+1)), \qquad k \ge 0. \tag{12}$$

The purpose of this modification is to speed convergence. The algorithm remains convergent to some $r(\infty) \in \mathbb{R}^q$, but it is no longer consistent. Bounds on the error $\|r(\infty) - r^*\|_M$ are obtained in [9, 4].

- One can change the error criterion. For example, consider instead the minimization of the mean-square "Bellman error",

$$\min_r \mathsf{E}_\pi[(Ph^r(X) - (1+\gamma)h^r(X) + c(X))^2]$$

Or, one might ask, why focus exclusively on this L_2 norm? The L_1 error may be more easily justified

$$\min_r \mathsf{E}_\pi[|Ph^r(X) - (1+\gamma)h^r(X) + c(X)|],$$

where in each case again $h^r(X) = r^{\mathrm{T}}\psi(X)$.

On differentiating we obtain a fixed point equation that can be solved using S.A. In the first the optimal parameter r^* satisfies,

$$\mathsf{E}_\pi[(r^{\mathrm{T}}(P\psi(X) - (1+\gamma)\psi(X) + c(X))(P\psi(X) - (1+\gamma)\psi(X))] = 0,$$

and in the second

$$\mathsf{E}_\pi[\mathrm{sign}\,[r^{\mathrm{T}}(P\psi(X) - (1+\gamma)\psi(X) + c(X)](P\psi(X) - (1+\gamma)\psi(X))] = 0.$$

The associated S.A. recursion appears to be complex since one must estimate $P\psi$.

- A simplification is obtained on eliminating the conditional expectation. Consider for simplicity the L_2 setting with,

$$\min_r \mathsf{E}_\pi[(h^r\,(X(k+1)) - (1+\gamma)h^r(X(k)) + c(X(k)))^2] \tag{13}$$

The minimization (13) is easily solved using S.A. since we don't have to estimate $P\psi$: The optimal parameter r^* satisfies,

$$\mathsf{E}_\pi[(r^{\mathrm{T}}(\psi(X(k+1)) - (1+\gamma)\psi(X(k)) + c(X(k)))(\psi(X(k+1)) - (1+\gamma)\psi(X(k)))] = 0.$$

This can be computed by simulating the deterministic O.D.E.,

$$\begin{aligned}\tfrac{d}{dr}r(t) &= -a\nabla_r \mathsf{E}_\pi[(h^r\,(X(k+1)) - (1+\gamma)h^r(X(k)) + c(X(k)))^2]\\ &= -a\mathsf{E}_\pi[(r^{\mathrm{T}}(t)(\psi(X(k+1)) - (1+\gamma)\psi(X(k)) + c(X(k)))(\psi(X(k+1)) - (1+\gamma)\psi(X(k)))].\end{aligned}$$

The associated discrete-time algorithm is similar to TD(λ),

$$r(k+1) - r(k) = a_k d(k) z(k+1), \qquad k \geq 0,$$

with $d(k)$ again defined in (10), and $z(k+1) := (1+\gamma)h^{r(k)}(X(k)) - h^{r(k)}(X(k+1))$.

- Finally, with an appropriate notion of distance, one can compute an optimal approximation h^{r^*} using a linear program (LP), or a simulation-based approximate LP [3].

References

[1] D. P. Bertsekas, V. Borkar, and A. Nedic. Improved temporal difference methods with linear function approximation. In J. Si, A. Barto, W. Powell, and D. Wunsch, editors, *Handbook of Learning and Approximate Dynamic Programming*, pages 690–705. Wiley-IEEE Press, Piscataway, NJ., 2004.

[2] D.P. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming.* Atena Scientific, Cambridge, Mass, 1996.

[3] D. P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Res.*, 51(6):850–865, 2003.

[4] M. Kearns and S. Singh. Bias-variance error bounds for temporal difference updates. In *Proceedings of the 13th Annual Conference on Computational Learning Theory*, pages 142–147, 2000.

[5] S. Mannor, I. Menache, and N. Shimkin. Basis function adaptation in temporal difference reinforcement learning. *Annals of Operations Res.*, 134(2):215–238, 2005.

[6] A. Nedic and D.P. Bertsekas. Least squares policy evaluation algorithms with linear function approximation. *Discrete Event Dynamic Systems: Theory and Applications*, 13(1-2):79–110, 2003.

[7] B. Van Roy. Neuro-dynamic programming: Overview and recent trends. In E. Feinberg and A. Shwartz, editors, *Markov Decision Processes: Models, Methods, Directions, and Open Problems*, pages 43–82. Kluwer, Holland, 2001.

[8] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* MIT Press (and on-line, http://www.cs.ualberta.ca/%7Esutton/book/ebook/the-book.html), 1998.

[9] J. N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Trans. Automat. Control*, 42(5):674–690, 1997.