Basis Function Adaptation Methods for Cost Approximation in MDP

Huizhen Yu
Department of Computer Science and HIIT
University of Helsinki
Helsinki 00014, Finland
Email: janey.yu@cs.helsinki.fi

Dimitri P. Bertsekas
Laboratory for Information and Decision Systems (LIDS)
Massachusetts Institute of Technology
MA 02139, USA
Email: dimitrib@mit.edu

***Abstract*—We generalize a basis adaptation method for cost approximation in Markov decision processes (MDP), extending earlier work of Menache, Mannor, and Shimkin. In our context, basis functions are parametrized and their parameters are tuned by minimizing an objective function involving the cost function approximation obtained when a temporal differences (TD) or other method is used. The adaptation scheme involves only low order calculations and can be implemented in a way analogous to policy gradient methods. In the generalized basis adaptation framework we provide extensions to TD methods for nonlinear optimal stopping problems and to alternative cost approximations beyond those based on TD.**

I. Overview

We consider a parameter optimization context consisting of a parameter vector $\theta \in \Theta$, where Θ is an open subset of $\Re^k$, a function $x^* : \Theta \mapsto \Re^n$, and a cost function $F : \Re^n \mapsto \Re$. We want to solve the problem

$$\min_{\theta \in \Theta} \; F\big(x^*(\theta)\big).$$

In general, for a given θ, the vector $x^*(\theta)$ may be the result of an algorithm, a design process, or the solution of an equation (all parametrized by θ), but in this paper we focus on a special context that arises commonly in approximate dynamic programming (ADP). The salient features of this context are:

- $x^*(\theta)$ is an approximation of the cost vector of an n-state Markovian decision problem (MDP) within a subspace
$$S_\theta = \big\{\Phi(\theta) r \mid r \in \Re^s\big\},$$
where the s columns of the $n \times s$ matrix Φ are basis functions parametrized by θ.
- n, the dimension of x as well as the number of states, is much larger than s, the dimension of the approximation subspace S_θ.

We then obtain a basis adaptation problem: optimally select the basis functions within a parametric class so that the cost vector approximation $x^*(\theta)$ has some desirable properties. This is related to the issue of selecting a good approximation architecture, a subject of substantial research interest at present in ADP. Note that the parameters in the basis functions may have a natural interpretation; for instance, they may reflect the degree of smoothness that we believe the cost function has, particularly when the problem is a discretized version of a continuous problem; or they may be parameters of a function that maps states to quantitative features used to generate the rows of Φ.

In this paper, we consider basis adaptation methods that use gradient-based optimization and low dimensional calculations (order s rather than n). We are motivated by two general settings, which involve some form of Bellman's equation, $x = T(x)$, corresponding to a single or to multiple policies.

In the first setting, S_θ is a general s-dimensional subset of $\Re^n$, and $x^* : \Theta \mapsto \Re^n$ is the fixed point of the mapping T, left composed by a "projection" mapping $\Pi(\cdot, \theta) : \Re^n \to S_\theta$ associated with θ, i.e., $x^*(\theta)$ is the solution of the "projected" equation

$$x = \Pi\big(T(x), \theta\big). \tag{1}$$

In the second setting, $x^*(\theta)$ is defined differently, as the solution of an optimization problem:

$$x^*(\theta) \in \arg\min_{x \in X(\theta)} f(x, \theta), \tag{2}$$

where for each θ, $f(\cdot, \theta)$ and $X(\theta)$ are suitable cost function and constraint, respectively. One example is the linear regression approach, where

$$f(x, \theta) = \|x - T(x)\|^2, \quad X(\theta) = \{\Phi(\theta) r \mid r \in \Re^s\}.$$

Another example is when $x^*(\theta)$ is obtained from an approximate linear programming formulation.

Most of the paper is devoted to the first setting where $x^*(\theta)$ is the solution of the projected equation (1). To illustrate our approach, let us assume for the time being that such a solution exists, is unique, and is differentiable in a neighborhood of a given θ; this can be implied from the differentiability of the mappings Π, T at certain points, and the appropriate implicit function theorems. Then, by differentiating both sides of Eq. (1), we obtain the equation satisfied by the partial derivatives of $x^*(\theta)$ with respect to the components θ_j of θ. In particular, for each j, the n-dimensional vector $\frac{\partial x^*}{\partial \theta_j}(\theta)$ satisfies the linear equation

$$\begin{aligned}\tfrac{\partial x^*}{\partial \theta_j}(\theta) = {} & \tfrac{\partial \Pi}{\partial \theta_j}\big(T(x^*), \theta\big) \\ & + \nabla_y \Pi(y, \theta)\big|_{y = T(x^*)} \cdot \nabla T(x^*) \cdot \tfrac{\partial x^*}{\partial \theta_j}(\theta),\end{aligned} \tag{3}$$

978-1-4244-2761-1/09/$25.00 ©2009 IEEE

where ∇ denotes the Jacobian of the associated mapping with respect to a corresponding argument. This equation (and hence also the values of $\frac{\partial x^*}{\partial \theta_j}(\theta)$) does not depend on how the points in S_θ are represented. However, given a general parametrization of the form

$$S_\theta = \{\psi(r,\theta) \mid r \in \Re^s\},$$

we have $x^*(\theta) = \psi(r^*,\theta)$ for some $r^* \in \Re^s$, so that

$$\frac{\partial x^*}{\partial \theta_j}(\theta) = \frac{\partial \psi}{\partial \theta_j}(r^*,\theta) + \nabla_r \psi(r^*,\theta) \cdot \frac{\partial r^*}{\partial \theta_j}(\theta) \qquad (4)$$

(assuming differentiability of the functions involved for the time being). Thus, after $x^*(\theta)$ (equivalently, $r^*(\theta)$) is computed, $\frac{\partial x^*}{\partial \theta_j}(\theta)$ can be efficiently represented without explicitly storing length-n vectors, and by substituting Eq. (4) in (3), solving Eq. (3) reduces to solving a low-dimensional linear equation in $\frac{\partial r^*}{\partial \theta_j}(\theta)$. This is important for computational reasons. Similarly, when all the mappings are twice differentiable, we can derive the equations satisfied by the second-order derivatives.

Suppose that we wish to optimize over Θ a certain objective function $F(x^*(\theta))$. Then we can use $\frac{\partial x^*}{\partial \theta_j}(\theta)$ computed as above and the chain rule to compute the gradient $\nabla(F \circ x^*)$, and apply gradient methods. In the context of TD-based approximate policy evaluation for a discounted MDP, this approach was first proposed by Menache, Mannor, and Shimkin [1], who gave a simulation-based gradient method that uses low-dimensional calculations for the case of the linear Bellman equation $x = T(x)$ corresponding to a single policy, and the Bellman error criterion

$$F\big(x^*(\theta)\big) = \big\|x^*(\theta) - T\big(x^*(\theta)\big)\big\|^2.$$

A central element of their approach is a convenient LSTD-type iterative algorithm for estimating the partial derivatives of x^* with respect to components of θ. Their analysis still relies on the availability of analytic expressions of x^* and r^* in the linear problems they consider. One main purpose of this paper is to show a broader scope of this adaptation methodology beyond linear problems, as we have put forth already in Eqs. (1)-(4). Our work generalizes the work of [1] so that it applies to alternative optimization criteria, including some involving a nonlinear Bellman equation (as for example in optimal stopping), and also applies to cost approximation by other methods, unrelated to the projected equation (1) (such as for example regression).

Generally, there are potential difficulties at two levels when using the method just outlined for basis adaptation:

- At the analytical level, issues such as (generalized) differentiability and expressions of the Jacobians can be highly complex and difficult to analyze.
- At the computational level, there can be difficulties in estimating the necessary quantities defining the equations and solving them using low-order simulation-based calculation, and there can also be difficult convergence issues.

In this paper, we will limit ourselves to the analytically simpler case where S_θ is a linear subspace, $S_\theta = \{\Phi(\theta)r \mid r \in \Re^s\}$, and $\Pi(\cdot,\theta)$ is a Euclidean projection, hence a linear mapping. Thus the projected equation simplifies to

$$x = \Pi(\theta)T(x),$$

and $x^*(\theta)$ is simply $\Phi(\theta)r^*(\theta)$. For $\Pi(\theta)$ to be differentiable at θ, it is sufficient that $\Phi(\theta)$ is differentiable with linearly independent columns. In Section II, T will be a linear mapping associated with a single policy, while in Section III, T will be a nonlinear mapping associated with an optimal stopping problem. In the latter case, T may be nondifferentiable, so we will use a more general differentiability notion in our analysis. There are also other nonlinear cases (e.g., constrained cases) that are relatively simple and are worthy of future investigation.

The paper is organized as follows. In Sections II we review the application of the basis adaptation scheme for policy evaluation with TD. We recount the case considered by [1], and we also include additional cases. In Section III, we consider optimal cost approximation in optimal stopping problems using TD(0). In this case, the mapping T is not differentiable, and as a result the solution $x^*(\theta)$ of the associated nonlinear projected Bellman equation may not be differentiable. It is instead semidifferentiable, (as we show in an Appendix to this paper, available online). A corresponding difficulty arises in estimating the semiderivatives/directional derivatives by simulation-based algorithms. We discuss a smoothing approach: applying basis adaptation to the solution of the projected version of a smoothed Bellman equation. In Section IV, as an example of the second setting, we discuss basis adaptation for the linear regression approach based on minimizing the Bellman equation error.

II. Review of a Case of Linear Equations: TD and Policy Evaluation

We consider the case of policy evaluation using TD(λ) with linear function approximation. TD(λ) was first proposed by Sutton [2]; for textbook discussions of TD(λ) and alternative TD methods, see [3]–[5]. The mapping T is the linear mapping associated with the multiple-step Bellman equation. In particular, for a given pair of (α,λ) with $\alpha \in (0,1), \lambda \in [0,1]$,

$$T(x) = g^{(\lambda)} + P^{(\lambda)}x$$

where $g^{(\lambda)}$ is a vector and $P^{(\lambda)}$ a matrix defined by

$$g^{(\lambda)} = (I - \lambda\alpha P)^{-1}g, \quad P^{(\lambda)} = (1-\lambda)\sum_{m=0}^{\infty}\lambda^m(\alpha P)^{m+1}, \qquad (5)$$

respectively, with P being the transition matrix of the Markov chain associated with the policy, g being the one-stage cost vector, and α being the discount factor. We consider the discounted case $\alpha \in (0,1)$ only for notational simplicity, and the discussion that follows extends easily to undiscounted cases.

For all $\theta \in \Theta$, the approximation space S_θ is the linear s-dimensional subspace of $\Re^n$ spanned by the columns of a matrix $\Phi(\theta)$. We assume that for all θ, $\Phi(\theta)$ is differentiable

and its columns are linearly independent. Let $\Pi(\theta)$ be a weighted Euclidean projection on S_θ, thus a linear mapping. We assume that for all θ, the matrix $I-\Pi(\theta)P^{(\lambda)}$ is invertible.

Then, for all $\theta \in \Theta$, there exists a unique solution $x^*(\theta)$ to the projected Bellman equation associated with TD(λ),

$$x = \Pi(\theta)T(x) = \Pi(\theta)(g^{(\lambda)} + P^{(\lambda)}x),$$

and $x^*(\theta)$ is differentiable on Θ. Equation (3) becomes for $j = 1, \ldots, k$,

$$\frac{\partial x^*}{\partial \theta_j}(\theta) = \frac{\partial \Pi}{\partial \theta_j}(\theta)T(x^*) + \Pi(\theta)P^{(\lambda)}\frac{\partial x^*}{\partial \theta_j}(\theta). \tag{6}$$

Equation (4) becomes

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

with $\Phi(\theta)r^*(\theta) = x^*(\theta)$, ($r^*(\theta)$ is differentiable since $\Phi(\theta)$ is differentiable and has linearly independent columns). It follows from combining the above two equations that the second component $\Phi(\theta)\frac{\partial r^*}{\partial \theta_j}(\theta)$ of the derivative is the solution of the equation

$$x = q_j(x^*) + \Pi(\theta)P^{(\lambda)}x \tag{7}$$

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

$$q_j(x^*) = \frac{\partial \Pi}{\partial \theta_j}(\theta)T(x^*) + (\Pi(\theta)P^{(\lambda)} - I)\frac{\partial \Phi}{\partial \theta_j}(\theta)r^*. \tag{8}$$

(Note that since r^* in this case has an explicit expression, alternatively, we can differentiate the expression of r^* directly to get an identical formula for derivative estimation.)

We can use various TD algorithms to solve Eq. (7), as we will discuss shortly. In a way this is similar to actor-critic methods (see e.g., [6]): the gradient satisfies linear equations [(7) for all j] of the same form as the equation satisfied by the approximating cost x^*, except that the constant terms of these equations are defined through x^*. For gradient estimation, least squares-based algorithms, such as LSTD [7] and LSPE [8] are particularly convenient, because $q_j(x^*)$ is also linear in r^*, so the terms multiplying r^* in its expression can be estimated simultaneously as r^* itself is being estimated. In what follows, we describe the details, starting with expressing Eq. (7) explicitly in terms of low dimensional quantities.

A. Projection Norm Independent of θ

Let ξ be the weights in the projection norm, and let Ξ denote the diagonal matrix with diagonal entries being ξ. We first consider the case where ξ does not functionally depend on θ. The projection $\Pi(\theta)$ can be expressed in matrix notation as

$$\Pi(\theta) = \Phi(\theta)(\Phi(\theta)'\,\Xi\,\Phi(\theta))^{-1}\Phi(\theta)'\,\Xi.$$

To simplify notation, define matrices B_0 and B_j, $j = 1, \ldots, k$, by

$$B_0(\theta) = \Phi(\theta)'\,\Xi\,\Phi(\theta), \quad B_j(\theta) = \frac{\partial \Phi}{\partial \theta_j}(\theta)'\,\Xi\,\Phi(\theta), \tag{9}$$

and also omit θ in the parentheses for the time being. A useful fact is that for any invertible square matrix B differentiable in θ, $\frac{\partial B^{-1}}{\partial \theta_j} = -B^{-1}\frac{\partial B}{\partial \theta_j}B^{-1}$. It can be easily verified that

$$\frac{\partial \Pi}{\partial \theta_j} = \frac{\partial \Phi}{\partial \theta_j}B_0^{-1}\Phi'\,\Xi - \Phi B_0^{-1}\big(B_j + B_j'\big)B_0^{-1}\Phi'\,\Xi + \Phi B_0^{-1}\frac{\partial \Phi}{\partial \theta_j}'\,\Xi. \tag{10}$$

Substituting this expression in Eq. (8), using the fact $B_0^{-1}\Phi'\,\Xi\,T(x^*) = r^*$, and rearranging terms, we can express $q_j(x^*)$ explicitly in terms of low-dimensional quantities:

$$q_j(x^*) = \Phi\hat{r}_j, \text{ with } \hat{r}_j = B_0^{-1}\left(\frac{\partial \Phi}{\partial \theta_j}'\,\Xi\, g^{(\lambda)} + M_j r^*\right), \tag{11}$$

where

$$M_j = \frac{\partial \Phi}{\partial \theta_j}'\,\Xi\,(P^{(\lambda)} - I)\Phi + \Phi'\,\Xi\,(P^{(\lambda)} - I)\frac{\partial \Phi}{\partial \theta_j}. \tag{12}$$

We can also write Eq. (7) equivalently as

$$M_0 r + B_0\hat{r}_j = 0$$

where

$$M_0 = \Phi'\,\Xi(P^{(\lambda)} - I)\Phi. \tag{13}$$

B. Projection Norm Dependent on θ

We now consider the case where ξ depends on θ. The expression of $\frac{\partial \Pi}{\partial \theta_j}(\theta)$ contains two more terms in addition to those in Eq. (10):

$$-\Phi B_0^{-1}\big(\Phi'\frac{\partial \Xi}{\partial \theta_j}\Phi\big)B_0^{-1}\Phi'\,\Xi, \quad \text{and} \quad \Phi B_0^{-1}\Phi'\,\frac{\partial \Xi}{\partial \theta_j}.$$

Correspondingly, it is easy to verify that now $q_j(x^*) = \Phi\hat{r}_j$ with

$$\hat{r}_j = B_0^{-1}\left(\frac{\partial \Phi}{\partial \theta_j}'\,\Xi\, g^{(\lambda)} + M_j r^* + \Phi'\,\frac{\partial \Xi}{\partial \theta_j}g^{(\lambda)} + \widehat{M}_j r^*\right), \tag{14}$$

where M_j is given by Eq. (12), and

$$\widehat{M}_j = \Phi'\frac{\partial \Xi}{\partial \theta_j}(P^{(\lambda)} - I)\Phi. \tag{15}$$

Since the expressions involve $\frac{\partial \xi}{\partial \theta_j}$, this setting is realistic only when $\xi(\theta)$ is known explicitly. It is more suitable for TD(0) but can be difficult to apply for TD(λ) with $\lambda > 0$ (see related details in Example 4 below).

C. Examples of Derivative Estimation

We define some notation to be used throughout the paper except where stated otherwise. We denote by $g(i, i')$ the one-stage cost of transition from state i to i', and we denote by $\phi(i)$ the ith row of Φ, viewed as a column vector. Let $\{\gamma_t\}$ denote a deterministic sequence of positive stepsizes satisfying the standard condition: $\sum_{t\geq 0}\gamma_t = \infty$, $\sum_{t\geq 0}\gamma_t^2 < \infty$. Let $(i_0, i_1, \ldots)$ be a sample trajectory of states from an irreducible Markov chain with invariant distribution ξ. This Markov chain is assumed to be the Markov chain associated with the policy, i.e., it has as transition matrix P, except where noted otherwise.

Example 1 (ξ independent of θ)**.** This is the case considered in [1]. The component $\Phi\frac{\partial r^*}{\partial \theta_j}$ of the derivative $\frac{\partial x^*}{\partial \theta_j}$ satisfies

Eq. (7) with $q_j = \Phi \hat{r}_j$ given by Eq. (11). To estimate $\frac{\partial x^*}{\partial \theta_j}$, we define vector iterates

$$z_{0,t} = \lambda\alpha z_{0,t-1} + \phi(i_t), \quad z_{j,t} = \lambda\alpha z_{j,t-1} + \tfrac{\partial \phi(i_t)}{\partial \theta_j},$$
$$b_{0,t} = (1-\gamma_t) b_{0,t-1} + \gamma_t z_{0,t} g(i_t, i_{t+1}),$$
$$b_{j,t} = (1-\gamma_t) b_{j,t-1} + \gamma_t z_{j,t} g(i_t, i_{t+1}),$$

and matrix iterates

$$B_{0,t} = (1-\gamma_t) B_{0,t-1} + \gamma_t \phi(i_t)\phi(i_t)',$$
$$M_{0,t} = (1-\gamma_t) M_{0,t-1} + \gamma_t z_{0,t} (\alpha\phi(i_{t+1}) - \phi(i_t))',$$
$$M_{j,t} = (1-\gamma_t) M_{j,t-1} + \gamma_t z_{j,t} (\alpha\phi(i_{t+1}) - \phi(i_t))' + \gamma_t z_{0,t} \left(\alpha \tfrac{\partial \phi(i_{t+1})}{\partial \theta_j} - \tfrac{\partial \phi(i_t)}{\partial \theta_j}\right)'.$$

It can be shown that as $t \to \infty$, $B_{0,t} \to B_0$, $M_{0,t} \to M_0$, $M_{j,t} \to M_j$, $b_{0,t} \to \Phi' \Xi\, g^{(\lambda)}$, and $b_{j,t} \to \frac{\partial \Phi}{\partial \theta_j} \Xi\, g^{(\lambda)}$, with probability 1 (w.p.1). We then let $r_{0,t}$ and $r_{j,t}$ be defined either by the LSTD algorithm:

$$r_{0,t+1} = -M_{0,t}^{-1} b_{0,t},$$
$$r_{j,t+1} = -M_{0,t}^{-1} (b_{j,t} + M_{j,t} r_{0,t});$$

or, by the LSPE algorithm with a constant stepsize γ (e.g, $\gamma \in (0,1]$):

$$r_{0,t+1} = r_{0,t} + \gamma B_{0,t}^{-1} (M_{0,t} r_{0,t} + b_{0,t}),$$
$$r_{j,t+1} = r_{j,t} + \gamma B_{0,t}^{-1} (M_{0,t} r_{j,t} + b_{j,t} + M_{j,t} r_{0,t}).$$

It follows from the convergence of LSTD/LSPE that w.p.1, $\lim_{t\to\infty} r_{0,t} = r^*$ and

$$\lim_{t\to\infty} r_{j,t} = \tfrac{\partial r^*}{\partial \theta_j}, \qquad \lim_{t\to\infty} \tfrac{\partial \Phi}{\partial \theta_j} r_{0,t} + \Phi r_{j,t} = \tfrac{\partial x^*}{\partial \theta_j}. \quad \square$$

Example 2. We continue with the above example. Suppose the objective function $F(x)$ for basis adaptation is the Bellman error:

$$F(x^*) = \tfrac{1}{2} \|x^* - g - \alpha P x^*\|_\xi^2.$$

Then, $\nabla(F \circ x^*)(\theta) = (\ldots, \frac{\partial (F \circ x^*)}{\partial \theta_j}(\theta), \ldots)$ with

$$\tfrac{\partial (F \circ x^*)}{\partial \theta_j} = \big\langle x^* - g - \alpha P x^*,\ \tfrac{\partial x^*}{\partial \theta_j} - \alpha P \tfrac{\partial x^*}{\partial \theta_j} \big\rangle_\xi,$$

where $< \cdot, \cdot >_\xi$ denotes inner product and we omit θ in the parentheses for simplicity. Estimation using samples can be done efficiently by approximating the matrices and vectors which do not depend on x^* and plugging in the approximations of r^* and $\frac{\partial r^*}{\partial \theta_j}$ as they are computed. In particular, we have

$$\tfrac{\partial (F \circ x^*)}{\partial \theta_j} = r^{*\prime} \Phi' (I - \alpha P)' \Xi (I - \alpha P) \big(\tfrac{\partial \Phi}{\partial \theta_j} r^* + \Phi \tfrac{\partial r^*}{\partial \theta_j}\big) - g' \Xi (I - \alpha P) \big(\tfrac{\partial \Phi}{\partial \theta_j} r^* + \Phi \tfrac{\partial r^*}{\partial \theta_j}\big). \tag{16}$$

To compute this quantity, for state i_t along a trajectory of states $(i_0, i_1, \ldots)$, we sample an additional transition $\bar{i}_{t+1}$ from i_t, independently of the trajectory, and do the matrix and vector iterates:

$$V_{j,t} = (1-\gamma_t) V_{j,t-1} + \gamma_t \left(\phi(i_t) - \alpha\phi(i_{t+1})\right) \begin{bmatrix} \frac{\partial \phi(i_t)}{\partial \theta_j} - \alpha \frac{\partial \phi(\bar{i}_{t+1})}{\partial \theta_j} \\ \phi(i_t) - \alpha \phi(\bar{i}_{t+1}) \end{bmatrix}',$$
$$v_{j,t} = (1-\gamma_t) v_{j,t-1} + \gamma_t g(i_t, i_{t+1}) \begin{bmatrix} \frac{\partial \phi(i_t)}{\partial \theta_j} - \alpha \frac{\partial \phi(\bar{i}_{t+1})}{\partial \theta_j} \\ \phi(i_t) - \alpha \phi(\bar{i}_{t+1}) \end{bmatrix}'.$$

Then, as $t \to \infty$, w.p.1,

$$V_{j,t} \to \Phi'(I - \alpha P)' \Xi (I - \alpha P) \left[\tfrac{\partial \Phi}{\partial \theta_j}\ \Phi\right],$$
$$v_{j,t} \to g' \Xi (I - \alpha P) \left[\tfrac{\partial \Phi}{\partial \theta_j}\ \Phi\right],$$

so, with $r_{0,t}$ and $r_{j,t}$ given by Example 1,

$$r_{0,t}' V_{j,t} \begin{bmatrix} r_{0,t} \\ r_{j,t} \end{bmatrix} - v_{j,t} \begin{bmatrix} r_{0,t} \\ r_{j,t} \end{bmatrix} \to \tfrac{\partial (F \circ x^*)}{\partial \theta_j}, \quad w.p.1.$$

One may also consider a two-time-scale algorithm which estimates the gradient at a faster time-scale, by using the iterates given in this and the preceding example (with $\gamma_t = t^{-\beta}, \beta \in (\frac{1}{2}, 1)$ for instance), and changes θ_t at a slower time-scale along the estimated gradient direction. Then it is not difficult to show, under standard conditions, that w.p.1, θ_t converges to the closure of the set $\{\theta \mid \nabla(F \circ x^*)(\theta) = 0, \theta \in \Theta\}$, by using e.g., the results in Borkar [9] and [10]. $\square$

Example 3. As an example of finding θ to minimize an objective function different than the Bellman error, we may consider

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

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. We may use this criterion to tune the subspace S_θ, while we use TD to obtain approximating costs at the rest of states. The gradient $\nabla(F \circ x^*)$ can be easily calculated from the estimates of ∇x^* given in Example 1. $\square$

Example 4 (ξ dependent on θ)**.** We assume that for all states i, the ratios $\frac{\partial \xi(i)}{\partial \theta_j}(\theta)/\xi(i)$ are known, and ξ is the invariant distribution of the Markov chain that we are simulating, whose transition matrix is $\widehat{P}$. We also assume that the ratios $\frac{p_{ii'}}{\hat{p}_{ii'}}$ between the entries of P and $\widehat{P}$ are well-defined and known.

The component $\Phi \frac{\partial r^*}{\partial \theta_j}$ of the derivative $\frac{\partial x^*}{\partial \theta_j}$ satisfies Eq. (7) with $q_j = \Phi \hat{r}_j$ now given by Eq. (14). To estimate $\frac{\partial x^*}{\partial \theta_j}$, we use weighted sampling to handle the terms in the expression of $\hat{r}_j$ that involve $\Phi' \frac{\partial \Xi}{\partial \theta_j}$ based on the observation that we can write for $j = 1, \ldots, k$,

$$\Phi' \tfrac{\partial \Xi}{\partial \theta_j} = \begin{bmatrix} w(1)\phi(1) & \cdots & w(n)\phi(n) \end{bmatrix} \cdot \Xi, \tag{17}$$

where $w(i) = \frac{\partial \xi(i)}{\partial \theta_j} / \xi(i), i = 1, \ldots, n$. From this it is easy to see how to estimate the two related terms $\Phi' \frac{\partial \Xi}{\partial \theta_j} g^{(\lambda)}$ and $\widehat{M}_j$ in Eq. (14) using standard TD formulas. For estimating

terms involving $P^{(\lambda)}$, since the states are sampled using a Markov chain with transition matrix $\widehat{P}$ instead of P, we employ weighting of samples to account for this discrepancy (see [11]).

In particular, similar to Example 1, we define the following vector iterates

$$z_{0,t} = \lambda\alpha \tfrac{p_{i_{t-1}i_t}}{\hat{p}_{i_{t-1}i_t}} \cdot z_{0,t-1} + \phi(i_t),$$
$$z_{j,t} = \lambda\alpha \tfrac{p_{i_{t-1}i_t}}{\hat{p}_{i_{t-1}i_t}} \cdot z_{j,t-1} + \tfrac{\partial\phi(i_t)}{\partial\theta_j},$$
$$b_{0,t} = (1-\gamma_t)b_{0,t-1} + \gamma_t z_{0,t} g(i_t, i_{t+1}),$$
$$b_{j,t} = (1-\gamma_t)b_{j,t-1} + \gamma_t z_{j,t} g(i_t, i_{t+1}),$$

and matrix iterates

$$B_{0,t} = (1-\gamma_t)B_{0,t-1} + \gamma_t \phi(i_t)\phi(i_t)',$$
$$M_{0,t} = (1-\gamma_t)M_{0,t-1} + \gamma_t z_{0,t}\Big(\alpha \tfrac{p_{i_t i_{t+1}}}{\hat{p}_{i_t i_{t+1}}} \cdot \phi(i_{t+1}) - \phi(i_t)\Big)',$$
$$M_{j,t} = (1-\gamma_t)M_{j,t-1} + \gamma_t z_{j,t}\left(\alpha \tfrac{p_{i_t i_{t+1}}}{\hat{p}_{i_t i_{t+1}}} \cdot \phi(i_{t+1}) - \phi(i_t)\right)' + \gamma_t z_{0,t}\left(\alpha \tfrac{p_{i_t i_{t+1}}}{\hat{p}_{i_t i_{t+1}}} \cdot \tfrac{\partial\phi(i_{t+1})}{\partial\theta_j} - \tfrac{\partial\phi(i_t)}{\partial\theta_j}\right)'.$$

In addition, we define the following vector and matrix iterates

$$\hat{z}_{j,t} = \lambda\alpha \tfrac{p_{i_{t-1}i_t}}{\hat{p}_{i_{t-1}i_t}} \cdot \hat{z}_{j,t-1} + \phi(i_t) \cdot \tfrac{\partial\xi(i_t)}{\partial\theta_j}/\xi(i_t),$$
$$\hat{b}_{j,t} = (1-\gamma_t)\hat{b}_{j,t-1} + \gamma_t \hat{z}_{j,t} g(i_t, i_{t+1}),$$
$$\widehat{M}_{j,t} = (1-\gamma_t)\widehat{M}_{j,t-1} + \gamma_t \hat{z}_{j,t}\Big(\alpha \tfrac{p_{i_t i_{t+1}}}{\hat{p}_{i_t i_{t+1}}} \cdot \phi(i_{t+1}) - \phi(i_t)\Big)'.$$

We then define iterates $r_{0,t}$, $r_{j,t}$ by the LSTD algorithm:

$$r_{0,t+1} = -M_{0,t}^{-1} b_{0,t},$$
$$r_{j,t+1} = -M_{0,t}^{-1}\left(b_{j,t} + M_{j,t} r_{0,t} + \hat{b}_{j,t} + \widehat{M}_{j,t} r_{0,t}\right).$$

Then, it can be shown that $M_{0,t} \to M_0$, $M_{j,t} \to M_j$, and $\widehat{M}_{j,t} \to \widehat{M}_j$ in probability [cf. Eqs. (13), (12), (15), respectively]; and $b_{0,t} \to \Phi'\,\Xi\, g^{(\lambda)}$, $b_{j,t} \to \frac{\partial\Phi}{\partial\theta_j}\,\Xi\, g^{(\lambda)}$, and $\hat{b}_{j,t} \to \Phi' \frac{\partial\Xi}{\partial\theta_j} g^{(\lambda)}$ in probability [cf. Eq. (14)]. Consequently, it can shown that $r_{0,t} \to r^*$, $r_{j,t} \to \frac{\partial r^*}{\partial\theta_j}$, and $\frac{\partial\Phi}{\partial\theta_j} r_{0,t} + \Phi r_{j,t} \to \frac{\partial x^*}{\partial\theta_j}$ in probability. □

D. Estimation of Second-Order Derivatives

When Φ is twice differentiable in θ, x^* is also, and we can compute the second-order derivatives. In particular,

$$\tfrac{\partial^2 x^*}{\partial\theta_j\partial\theta_k} = \tfrac{\partial^2\Phi}{\partial\theta_j\partial\theta_k} r^* + \tfrac{\partial\Phi}{\partial\theta_j}\tfrac{\partial r^*}{\partial\theta_k} + \tfrac{\partial\Phi}{\partial\theta_k}\tfrac{\partial r^*}{\partial\theta_j} + \Phi\,\tfrac{\partial^2 r^*}{\partial\theta_j\partial\theta_k}, \quad (18)$$

where the first three terms can be obtained in the process of computing r^* and its first-order derivatives, and the last term $\Phi\,\frac{\partial^2 r^*}{\partial\theta_j\partial\theta_k}$ can be computed by solving a linear equation of the form

$$x = q_{jk} + \Pi(\theta)P^{(\lambda)}x$$

for some $q_{jk} \in S_\theta$. This equation is obtained by differentiating both sides of Eq. (7) at $\Phi\frac{\partial r^*}{\partial\theta_j}$ by θ_k, similar to the derivation of first-order derivatives. Estimating $\frac{\partial^2 x^*}{\partial\theta_j\partial\theta_k}$ and $\frac{\partial^2(F\circ x^*)}{\partial\theta_j\partial\theta_k}$ can also be done efficiently by simulation, similar to the examples in the preceding section. These quantities are useful in applying Newton's method to optimize the parameter θ.

III. A Case of Nonlinear Equations: TD and Optimal Stopping Problems

We consider optimal stopping problems and approximation of the optimal cost functions using TD(0). The mapping T is the nonlinear mapping associated with the Bellman equation:

$$T(x) = g + Af(x)$$

where $f(x) = \min\{c, x\}$, or, written component-wise,

$$f(x) = (f_1(x_1), \ldots, f_n(x_n)), \quad \text{where} \quad f_i(x_i) = \min\{c_i, x_i\},$$

with c_i being the cost of stopping at state i, and $A = \alpha P, \alpha \in (0,1)$, with P being the transition matrix of the Markov chain and α being the discount factor of the stopping problem. We consider the discounted case for simplicity, even though the discussion extends to the total cost case.

As in the previous section, for all $\theta \in \Theta$, the approximation space S_θ is a linear s-dimensional subspace of $\Re^n$ spanned by the columns of $\Phi(\theta)$, and the projection $\Pi(\theta)$ on S_θ is a weighted Euclidean projection with respect to the norm $\|\cdot\|_\xi$ where ξ denotes the weight vector. For simplicity, we assume in this section that ξ corresponds to the invariant distribution of the Markov chain. We also assume that for all θ, $\Phi(\theta)$ is differentiable and its columns are linearly independent. The approximating cost $x^*(\theta)$ is the unique solution of the projected Bellman equation

$$x = \Pi(\theta)T(x) = \Pi(\theta)\big(g + Af(x)\big). \quad (19)$$

Note that with the above choice of ξ, $\Pi(\theta)T$ is a contraction with $\|\Pi(\theta)T\|_\xi \le \alpha$ for all $\theta \in \Theta$.

In this case, x^* does not have an analytical expression, but it can be efficiently computed [12]–[14], thanks to the contraction property of $\Pi(\theta)T$. We will specialize the general basis adaptation method discussed in Section I to this case. Note, however, that T is not differentiable everywhere.

First we derive the derivative formula and an estimation algorithm for points at which T is differentiable. We then discuss the semidifferentiability of $x^*(\theta)$ (with some analysis deferred to the Appendix, available online), and an associated difficulty in semiderivative/directional derivative estimation due to the non-smoothness of the Bellman operator. We then present a smoothing approach to basis adaptation based on using a smoothed version of the Bellman equation.

A. Derivatives at Certain Differentiable Points

Consider a point $\theta \in \Theta$ such that for all states i,

$$x_i^*(\theta) \ne c_i. \quad (20)$$

Let $\delta[\ldots]$ denote the indicator function. We have

$$\nabla T(x^*) = A\nabla f(x^*),$$

where $\nabla f(x^*)$ is a diagonal matrix with the ith diagonal entry being

$$\tfrac{df_i}{dx_i}(x_i^*) = \delta[x_i^* < c_i]. \quad (21)$$

Since $x - \Pi(\theta)T(x)$ is differentiable at (x^*, θ) and the matrix $I - \Pi(\theta)A\nabla f(x^*)$ is non-singular (which can be seen from

the fact that the mapping $L(x) = \Pi(\theta)A\nabla f(x^*)x$ is also a contraction with respect to $\|\cdot\|_\xi$), by the implicit function theorem $x^*(\theta)$ is differentiable in a neighborhood of θ.

Specializing Eq. (3) of Section I to this case, we have

$$\frac{\partial x^*}{\partial \theta_j}(\theta) = \frac{\partial \Pi}{\partial \theta_j}(\theta)T(x^*) + \Pi(\theta)\, A\, \nabla f(x^*)\frac{\partial x^*}{\partial \theta_j}(\theta). \qquad (22)$$

We also have $\frac{\partial x^*}{\partial \theta_j}(\theta) = \frac{\partial \Phi}{\partial \theta_j}(\theta)\, r^*(\theta) + \Phi(\theta)\, \frac{\partial r^*}{\partial \theta_j}(\theta)$. It can be verified, similar to the derivation in the previous section, that the second component $\Phi(\theta)\, \frac{\partial r^*}{\partial \theta_j}(\theta)$ is the solution of the linear equation

$$x = q_j(x^*) + \Pi(\theta)\, A\, \nabla f(x^*)x \qquad (23)$$

with $q_j(x^*) \in S_\theta$ given as follows. Omitting θ in the parentheses for simplicity,

$$q_j(x^*) = \Phi \hat{r}_j$$

with

$$\begin{aligned}\hat{r}_j =& B_0^{-1}\left(\tfrac{\partial \Phi}{\partial \theta_j}{}'\,\Xi\, g + M_j r^*\right)\\ &+ B_0^{-1}\left(\tfrac{\partial \Phi}{\partial \theta_j}{}'\,\Xi\, A\big(f(x^*) - \nabla f(x^*)\,\Phi r^*\big)\right) \qquad (24)\end{aligned}$$

and

$$M_j = \tfrac{\partial \Phi}{\partial \theta_j}{}'\,\Xi\,(A\nabla f(x^*) - I)\Phi + \Phi'\,\Xi\,(A\nabla f(x^*) - I)\tfrac{\partial \Phi}{\partial \theta_j}. \qquad (25)$$

The following derivative estimation algorithm is then evident. Notice that in a neighborhood of x^*, $\nabla f(x)$ is constant, therefore continuous. This is important for the convergence of the simulation-based algorithms we use to estimate $\frac{\partial x^*}{\partial \theta_j}$.

Example 5. Let $(i_0, i_1, \ldots)$ be a sample trajectory of states from the Markov chain. We assume the availability of a sequence $r_{0,t}$ converging to r^*, which can be obtained by a number of TD algorithms, e.g., the least squares Q-learning algorithm and its convergent variants [14], the fixed point Kalman filter algorithm [13], and the recursive TD(0) algorithm [12]. Let γ_t be a sequence of stepsizes such that $\sum_t \gamma_t = \infty, \sum_t \gamma_t^2 < \infty$. We define scalar $\kappa_t \in \{0, 1\}$ and vector iterates

$$\begin{aligned}\kappa_{t+1} &= \delta\big[\phi(i_{t+1})' r_{0,t} < c_{i_{t+1}}\big],\\ b_{j,t} &= (1-\gamma_t)b_{j,t} + \gamma_t \tfrac{\partial \phi(i_t)}{\partial \theta_j} g(i_t, i_{t+1}),\\ b^s_{j,t} &= (1-\gamma_t)b^s_{j,t-1} + \gamma_t \alpha\, c_{i_{t+1}}\,(1-\kappa_{t+1})\,\tfrac{\partial \phi(i_t)}{\partial \theta_j},\end{aligned}$$

and define matrix iterates

$$\begin{aligned}M^s_{0,t} &= (1-\gamma_t)M^s_{0,t-1} + \gamma_t \phi(i_t)\,(\alpha\,\kappa_{t+1}\phi(i_{t+1}) - \phi(i_t))',\\ M^s_{j,t} &= (1-\gamma_t)M^s_{j,t-1} + \gamma_t \tfrac{\partial \phi(i_t)}{\partial \theta_j}\,(\alpha\,\kappa_{t+1}\phi(i_{t+1}) - \phi(i_t))'\\ &\quad + \gamma_t \phi(i_t)\left(\alpha\,\kappa_{t+1}\tfrac{\partial \phi(i_{t+1})}{\partial \theta_j} - \tfrac{\partial \phi(i_t)}{\partial \theta_j}\right)'.\end{aligned}$$

We define $r_{j,t}$ either by the LSTD algorithm:

$$r_{j,t+1} = -(M^s_{0,t})^{-1}\left(b_{j,t} + b^s_{j,t} + M^s_{j,t} r_{0,t}\right);$$

or, by the LSPE algorithm with a constant stepsize $\gamma \in (0, \frac{2}{1+\alpha})$ ($\gamma = 1$, for instance):

$$\begin{aligned}r_{j,t+1} = r_{j,t} &+ \gamma B_{0,t}^{-1}\left(M^s_{0,t} r_{j,t} + b_{j,t}\right)\\ &+ \gamma B_{0,t}^{-1}\left(b^s_{j,t} + M^s_{j,t} r_{0,t}\right),\end{aligned}$$

where $B_{0,t}$ is defined as in Example 1. □

Proposition 1. *Assume that θ is such that Eq. (20) holds, and that $r_{0,t} \to r^*$ as $t \to \infty$, w.p.*1*. For $j = 1, \ldots, k$, let $r_{j,t}$ be given by Example 5; then, w.p.*1*,*

$$\lim_{t\to\infty} r_{j,t} = \frac{\partial r^*}{\partial \theta_j}, \qquad \lim_{t\to\infty} \tfrac{\partial \Phi}{\partial \theta_j} r_{0,t} + \Phi r_{j,t} = \frac{\partial x^*}{\partial \theta_j}.$$

Proof: (Sketch) Since $\nabla f(x)$ is constant in a neighborhood of x^* and $r_{0,t} \to r^*$ as $t \to \infty$ w.p.1, $\kappa_{t+1} = \frac{df_{i_{t+1}}}{dx_{i_{t+1}}}(x^*_{i_{t+1}})$ for t sufficiently large, w.p.1. Then it can be seen that the iterates converge to their respective limits: $b_{j,t} \to \frac{\partial \Phi}{\partial \theta_j}{}'\,\Xi\, g$, $b^s_{j,t} \to \frac{\partial \Phi}{\partial \theta_j}{}'\,\Xi\, A\big(f(x^*) - \nabla f(x^*)\,\Phi r^*\big)$, [cf. Eq. (24)], $M^s_{0,t} \to \Phi'\,\Xi\,(A\nabla f(x^*) - I)\Phi$ [cf. Eq. (23)], and $M^s_{j,t} \to M_j$ [cf. Eq. (25)]. The claimed convergence then follows from the convergence of LSTD/LSPE, where for LSPE we use also the fact that $L(x) = q_j + \Pi A \nabla f(x^*)x$ is a contraction mapping with respect to $\|\cdot\|_\xi$. ■

Example 6. Consider using the Bellman error as the objective function F for basis adaptation:

$$F(x^*) = \tfrac{1}{2}\|x^* - g - \alpha P f(x^*)\|^2_\xi.$$

We have

$$\begin{aligned}\tfrac{\partial (F\circ x^*)}{\partial \theta_j} &= \big\langle x^* - g - \alpha P f(x^*)\,,\, \tfrac{\partial x^*}{\partial \theta_j} - \alpha P\, \nabla f(x^*)\, \tfrac{\partial x^*}{\partial \theta_j}\big\rangle_\xi\\ &= \big\langle x^* - \bar{g} - \alpha P \nabla f(x^*)x^*\,,\, \tfrac{\partial x^*}{\partial \theta_j} - \alpha P\, \nabla f(x^*)\tfrac{\partial x^*}{\partial \theta_j}\big\rangle_\xi\end{aligned}$$

where

$$\bar{g} = g + \alpha P\big(f(x^*) - \nabla f(x^*)\, x^*\big).$$

Estimating this inner product by simulation can be done exactly as in Example 2, except for the following differences. We define

$$\bar{k}_{t+1} = \delta\big[\phi(\bar{i}_{t+1})' r_{0,t} < c_{\bar{i}_{t+1}}\big],$$

which equals $\frac{df_{\bar{i}_{t+1}}}{dx_{\bar{i}_{t+1}}}(x^*_{\bar{i}_{t+1}})$ for t sufficiently large, w.p.1. We replace $\frac{\partial \phi(\bar{i}_{t+1})}{\partial \theta_j}$ and $\phi(\bar{i}_{t+1})$ in both $V_{j,t}$ and $v_{j,t}$ by $\bar{\kappa}_{t+1}\frac{\partial \phi(\bar{i}_{t+1})}{\partial \theta_j}$ and $\bar{\kappa}_{t+1}\phi(\bar{i}_{t+1})$, respectively. We also replace the term $\phi(i_{t+1})$ in $V_{j,t}$ by $\kappa_{t+1}\phi(i_{t+1})$, and finally, in $v_{j,t}$, we replace the term $g(i_t, i_{t+1})$ by by $g(i_t, i_{t+1}) + \alpha(1 - \kappa_{t+1})c_{i_{t+1}}$. □

B. Semidifferentiability of x^*

For $\theta \in \Theta$ such that $x^*_i(\theta) = c_i$ for some i, $x^*(\theta)$ may not be differentiable at θ. Nevertheless, x^* is "well-behaved" in the sense that $x^*(\theta)$ is *semidifferentiable* on Θ, as we show in the Appendix to this paper (available online). Semidifferentiability is stronger than the one-sided directional differentiability; it implies the latter and the continuity of the directional derivatives in the direction (see [15]). While we will not go deeply into semiderivative estimation in the present

paper, we continue our discussion a little further to illustrate a difficulty, and to motivate the subsequent smoothing approach.

First, we introduce some notation. For a semidifferentiable function $F(x)$, we denote by $dF(x)(v)$ its semiderivative at x for a direction v (which coincides with the one-sided directional derivative at x for v); and for a semidifferentiable mapping $F(x)$, we denote its semiderivative at x for v by $DF(x)(v)$. Consider the functions $f_i(x_i) = \min\{c_i, x_i\}, i = 1, \ldots, n$, in our stopping problem. The semiderivative of f_i at x_i for a scalar v is

$$df_i(x_i)(v) = \hat{f}_i(v; x_i),$$

where $\hat{f}_i : \Re \times \Re \to \Re$ is given by

$$\hat{f}_i(v; x_i) = \begin{cases} v, & x_i < c_i, \\ 0, & x_i > c_i, \\ \min\{0, v\}, & x_i = c_i. \end{cases} \tag{26}$$

It can be shown that $x^*(\theta)$ is semidifferentiable on Θ (Prop. 3 in the Appendix). Denote the semiderivative of x^* at θ for a direction $v = (v_1, \ldots, v_k)$ of θ by $Dx^*(\theta)(v)$. Applying the chain rule, it can be shown that the semiderivative $Dx^*(\theta)(v)$ is the unique solution of the following nonlinear equation of x:

$$x = \sum_{j=1}^{k} v_j \, \tfrac{\partial \Pi}{\partial \theta_j}(\theta) T(x^*) + \Pi(\theta) A \hat{f}(x; x^*) \tag{27}$$

where $\hat{f}(x; x^*) = \left(\hat{f}_1(x_1; x_1^*), \ldots, \hat{f}_n(x_n; x_n^*) \right)'$ with $\hat{f}_i$ given by Eq. (26).

Similar to the differentiable cases in the earlier sections, Eq. (27) for the semiderivative $Dx^*(\theta)(v)$ can be rewritten explicitly in terms of $\Phi(\theta)$ and

$$\Psi_v(\theta) = \sum_{j=1}^{k} v_j \, \tfrac{\partial \Phi}{\partial \theta_j}(\theta).$$

In particular, (omitting θ in $\Phi(\theta)$ and $\Psi_v(\theta)$ for notational simplicity),

$$Dx^*(\theta)(v) = \Psi_v r^* + \Phi Dr^*(\theta)(v)$$

where the second component $\Phi Dr^*(\theta)(v)$ is the solution of the nonlinear equation of x:

$$x = \Phi \hat{r}_v(x^*) + \Pi(\theta) A \hat{f}\big(\Psi_v r^* + x \, ; x^*\big), \tag{28}$$

where the first term $\Phi \hat{r}_v(x^*)$ is given by

$$\begin{aligned} \Phi \hat{r}_v(x^*) &= D\Pi(\theta)(v) T(x^*) - \Psi_v r^* \\ &= \Phi B_0^{-1} \left(\Psi_v' \, \Xi \, T(x^*) \right) \\ &\quad - \Phi B_0^{-1} \left(\Psi_v' \, \Xi \, \Phi + \Phi' \, \Xi \, \Psi_v \right) r^*. \end{aligned} \tag{29}$$

If x^* were known *exactly*, then, since $\Pi A \hat{f}(x; \bar{x})$ for any fixed $\bar{x}$ is a contraction mapping for x, Eq. (28) can be solved by TD-type simulation-based algorithms [12]–[14], similar to solving the projected Bellman equation of the stopping problems. However, because with simulation x^* is only approached in the limit, and also *because $\hat{f}$ is discontinuous in its second argument*, we find it difficult to ensure the convergence of the simulation-based algorithm for solving Eq. (28), such as the ones used in Example 5. This leads us to consider an alternative smoothing approach to basis adaptation, presented in the next section.

In connection with the preceding discussion, we note that if T is smooth but $\Pi(\theta)$ is semidifferentiable, we will not have the above difficulty to compute the semiderivatives of x^* (in fact we only need to solve linear equations as before), because θ is always given. As it is natural to have parametrized basis functions that are not everywhere differentiable with respect to the parameters, the corresponding estimation scheme for the semiderivatives of $x^*(\theta)$ can be useful for basis adaptation in such cases.

C. Approximation by a Smoothed Problem

We consider smoothing the Bellman equation. For any positive ϵ, define function $h_\epsilon : \Re \to \Re$, which is a smoothed version of $\min\{0, a\}$, by

$$h_\epsilon(a) = \begin{cases} a - \epsilon \exp\left\{ \frac{a}{2\epsilon} \right\}, & a \le 0, \\ -\epsilon \exp\left\{ -\frac{a}{2\epsilon} \right\}, & a > 0. \end{cases} \tag{30}$$

Then, h_ϵ is monotonic, twice differentiable, and

$$\sup_{a \in \Re} |h_\epsilon(a) - \min\{0, a\}| \le \epsilon. \tag{31}$$

Define $f_\epsilon(x) = \big(f_{\epsilon,1}(x_1), \ldots, f_{\epsilon,n}(x_n)\big)$ to be a smoothed version of $f(\cdot) = \min\{c, \cdot\}$ by

$$f_{\epsilon,i}(x_i) = c_i + h_\epsilon(x_i - c_i), \tag{32}$$

and define T_ϵ to be a "smoothed" Bellman operator by

$$T_\epsilon(x) = g + A f_\epsilon(x). \tag{33}$$

Then, f_ϵ and T_ϵ are monotonic and twice differentiable, and furthermore, $\Pi(\theta) T_\epsilon(\cdot)$ is a contraction mapping with respect to $\|\cdot\|_\xi$ (since $|\frac{df_{\epsilon,i}}{dx_i}| < 1$ for all i). Consequently, the smoothed projected Bellman equation

$$x = \Pi(\theta) T_\epsilon(x)$$

has a unique solution, which we denote by

$$x_\epsilon^*(\theta) = \Phi(\theta) r_\epsilon^*(\theta),$$

and which can be computed by simulation using a number of TD algorithms (e.g., [12]–[14]). By Eq. (31), a worst case bound on the difference $x_\epsilon^* - x^*$ can be shown:

$$\|x_\epsilon^*(\theta) - x^*(\theta)\|_\xi \le \tfrac{\alpha\epsilon}{1-\alpha}. \tag{34}$$

The derivatives of x_ϵ^* with respect to θ can be estimated similarly as in Section III-A with $\nabla f_\epsilon(x_\epsilon^*)$ replacing $\nabla f(x^*)$ in Eqs. (22)-(25).

Example 7. Let $(i_0, i_1, \ldots)$ be a sample trajectory of states from the Markov chain. We define iterates $B_{0,t}$ and $b_{j,t}$ as in Example 5. We define scalars y_t, κ_t and vector $b^s_{j,t}$ by

$$\begin{aligned} y_{t+1} &= \phi(i_{t+1})' r_{0,t}, \quad \kappa_{t+1} = \tfrac{df_{\epsilon,i_{t+1}}}{dx_{i_{t+1}}}\big(y_{t+1}\big), \\ b^s_{j,t} &= (1-\gamma_t) b^s_{j,t-1} \\ &\quad + \gamma_t \alpha \tfrac{\partial \phi(i_t)}{\partial \theta_j} \left(f_{\epsilon,i_{t+1}}(y_{t+1}) - \kappa_{t+1} y_{t+1} \right), \end{aligned}$$

and we define matrix iterates

$$\begin{aligned} M^s_{0,t} &= (1-\gamma_t) M^s_{0,t-1} + \gamma_t \phi(i_t) \big(\alpha \kappa_{t+1} \phi(i_{t+1}) - \phi(i_t) \big)', \\ M^s_{j,t} &= (1-\gamma_t) M^s_{j,t-1} + \gamma_t \tfrac{\partial \phi(i_t)}{\partial \theta_j} \big(\alpha \kappa_{t+1} \phi(i_{t+1}) - \phi(i_t) \big)' \\ &\quad + \gamma_t \phi(i_t) \Big(\alpha \kappa_{t+1} \tfrac{\partial \phi(i_{t+1})}{\partial \theta_j} - \tfrac{\partial \phi(i_t)}{\partial \theta_j} \Big)'. \end{aligned}$$

We then define $r_{j,t}$ as in Example 5. □

Proposition 2. *Assume that $r_{0,t} \to r^*_\epsilon$ as $t \to \infty$, w.p.*1. *For $j = 1, \ldots, k$, let $r_{j,t}$ be given by Example 7; then, w.p.*1,

$$\lim_{t\to\infty} r_{j,t} = \tfrac{\partial r^*_\epsilon}{\partial \theta_j}, \qquad \lim_{t\to\infty} \tfrac{\partial \Phi}{\partial \theta_j} r_{0,t} + \Phi r_{j,t} = \tfrac{\partial x^*_\epsilon}{\partial \theta_j}.$$

Proof: (Sketch) Since $r_{0,t} \to r^*_\epsilon$ and $f_\epsilon, \nabla f_\epsilon$ are continuous, we have

$$f_{\epsilon,i_{t+1}}\big(\phi(i_{t+1})' r_{0,t}\big) = f_{\epsilon,i_{t+1}}\big(x^*_{\epsilon,i_{t+1}}\big) + o(1),$$

and

$$\kappa_{t+1} = \tfrac{df_{\epsilon,i_{t+1}}}{dx_{i_{t+1}}}\big(\phi(i_{t+1})' r_{0,t}\big) = \tfrac{df_{\epsilon,i_{t+1}}}{dx_{i_{t+1}}}\big(x^*_{\epsilon,i_{t+1}}\big) + o(1),$$

w.p.1, where $o(1)$ denotes some term that diminishes to 0 as $t \to \infty$. It then follows from e.g., Borkar [10], Chap. 2, Lemma 1 with its extension in Section 2.2, and Chap. 6, Corollary 8, that $b^s_{j,t}$, $M^s_{0,t}$, and $M^s_{j,t}$, $j \geq 1$, converge to their respective limits, whose expressions are as given in the proof of Prop. 1 with $\nabla f_\epsilon(x^*_\epsilon)$ replacing the term $\nabla f(x^*)$. The claimed convergence then follows. ■

IV. A Case of Parametric Optimization: Linear Regression for Policy Evaluation

Consider now

$$\min_{\theta\in\Theta} \min_{r\in\Re^s} F(r,\theta)$$

where

$$F(r,\theta) = \tfrac{1}{2} \|\Phi(\theta) r - g - A\Phi(\theta) r\|^2_\xi \tag{35}$$

with $A = \alpha P, \alpha \in (0,1)$, and $\Phi(\theta)$ is differentiable. Let

$$G(\theta) = \min_{r\in\Re^s} F(r,\theta). \tag{36}$$

Assume that for every θ there is a unique minimum r^*. Then, it follows from the theory of parametric minimization (or by a direct verification) that

$$\tfrac{\partial G}{\partial \theta_j}(\theta) = \tfrac{\partial F}{\partial \theta_j}(r^*, \theta), \tag{37}$$

which in our case, omitting θ in $\Phi(\theta)$ for simplicity, is

$$\tfrac{\partial G}{\partial \theta_j}(\theta) = \big\langle \Phi r^* - g - A\Phi r^*,\ \tfrac{\partial \Phi}{\partial \theta_j} r^* - A \tfrac{\partial \Phi}{\partial \theta_j} r^* \big\rangle_\xi. \tag{38}$$

The derivatives can be easily estimated by simulation.

Similarly, when $\Phi(\theta)$ is only semidifferentiable, derivatives in the above can be replaced by semiderivatives, and semiderivatives/directional derivatives of $G(\theta)$ can be estimated for basis adaptation.

V. Discussion

While we have considered primarily Markov decision problems in this paper, our basis selection scheme applies also in the context of approximately solving linear or nonlinear fixed point equations by TD methods [11] or by regression methods. For both contexts, choosing suitable objective functions F using prior knowledge about the problem can be beneficial, as the Bellman error or the residual error in satisfying the fixed point equation is not necessarily the best choice of the objective, especially when the problem is ill-conditioned.

Acknowledgment

We thank Prof. John Tsitsiklis, Prof. Paul Tseng, Prof. Vivek Borkar, and Nick Polydorides for helpful discussions. H. Yu is supported in part by Academy of Finland grant 118653 (ALGODAN) and by the IST Programme of the European Community IST-2002-506778 (PASCAL). D. P. Bertsekas is supported by NSF Grant ECCS-0801549.

References

[1] I. Menache, S. Mannor, and N. Shimkin, "Basis function adaptation in temporal difference reinforcement learning," *Ann. Oper. Res.*, vol. 134, no. 1, pp. 215–238, 2005.

[2] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine Learning*, vol. 3, pp. 9–44, 1988.

[3] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

[4] R. S. Sutton and A. G. Barto, *Reinforcement Learning*. Cambridge, MA: MIT Press, 1998.

[5] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Belmont, MA: Athena Scientific, 2007, vol. II.

[6] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," *SIAM J. Control Optim.*, vol. 42, no. 4, pp. 1143–1166, 2003.

[7] J. A. Boyan, "Least-squares temporal difference learning," in *Proc. The 16th Int. Conf. Machine Learning*, 1999.

[8] D. P. Bertsekas, V. S. Borkar, and A. Nedić, "Improved temporal difference methods with linear function approximation," in *Learning and Approximate Dynamic Programming*. IEEE Press, 2004.

[9] V. S. Borkar, "Stochastic approximation with 'controlled Markov' noise," *Systems Control Lett.*, vol. 55, pp. 139–145, 2006.

[10] ——, *Stochastic Approximation: A Dynamic Viewpoint*. New Delhi: Hindustan Book Agency, 2008.

[11] D. P. Bertsekas and H. Yu, "Projected equation methods for approximate solution of large linear systems," *J. Comput. Sci. Appl. Math.*, 2008, to appear.

[12] J. N. Tsitsiklis and B. Van Roy, "Optimal stopping of Markov processes: Hilbert space theory, approximation algorithms, and an application to pricing financial derivatives," *IEEE Trans. Automat. Contr.*, vol. 44, pp. 1840–1851, 1999.

[13] D. S. Choi and B. Van Roy, "A generalized Kalman filter for fixed point approximation and efficient temporal-difference learning," *Discrete Event Dyn. Syst.*, vol. 16, no. 2, pp. 207–239, 2006.

[14] H. Yu and D. P. Bertsekas, "A least squares Q-learning algorithm for optimal stopping problems," MIT, LIDS Tech. Report 2731, 2006.

[15] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*. Berlin: Springer-Verlag, 1998.

[16] S. M. Robinson, "An implicit-function theorem for a class of nonsmooth functions," *Math. Oper. Res.*, vol. 16, no. 2, pp. 292–309, 1991.

[17] A. L. Dontchev and R. T. Rockafellar, "Robinson's implicit function theorem and its extensions," *Math. Program. Ser. B*, vol. 117, no. 1, pp. 129–147, 2008.