A UNIFIED FRAMEWORK FOR LINEAR FUNCTION APPROXIMATION OF VALUE FUNCTIONS IN STOCHASTIC CONTROL

Matilde Sánchez-Fernández[*]

Universidad Carlos III de Madrid
Signal Theory & Communictions Dept.
Av. de La Universidad, 30
Leganés, 28911, Spain

Sergio Valcárcel, Santiago Zazo[†]

Universidad Politécnica de Madrid
Signals, Systems & Radiocommunications Dept.
Av. Complutense, 30
Madrid, 28040, Spain

ABSTRACT

This paper contributes with a unified formulation that merges previous analysis on the prediction of the performance (*value function*) of certain sequence of actions (*policy*) when an agent operates a Markov decision process with large state-space. When the states are represented by features and the value function is linearly approximated, our analysis reveals a new relationship between two common cost functions used to obtain the optimal approximation. In addition, this analysis allows us to propose an efficient adaptive algorithm that provides an unbiased linear estimate. The performance of the proposed algorithm is illustrated by simulation, showing competitive results when compared with the state-of-the-art solutions.

***Index Terms*—** Approximate dynamic programming, Linear value function approximation, Mean squared Bellman Error, Mean squared projected Bellman Error, Reinforcement Learning.

1. INTRODUCCTION

In communications and signal processing, many problems could be represented as Markov-decision-processes (MDP), in which an agent operates in an environment described by a set of states, the corresponding state-transition probabilities and some associated cost or rewards representing how much desirable for an agent is to be in each of the states [1, 2]. Examples of applications range from active monitoring problems (e.g., a wireless sensor networks aiming to keep a certain parameter in a certain range) to networking (e.g., call admission control, congestion avoidance and routing [3]). In these real world domains the number of possible states of the system is usually very large, turning the computation of the exact solution prohibitive; nevertheless an alternative formulation in terms of features that represent the states allows for efficient approximate solutions in the form of parametric representation of the value function. In particular, linear approximation [4, 1] has been the preferred option because of its simplicity, efficiency and good performance when the features are carefully chosen.

One of the benefits of using parametric approximation is that the problem can be posed as finding the optimal parameter that minimizes some cost function. Also it is of high interest how to implement the optimization of these cost functions guaranteeing fast and unbiased implementations. Several approaches in the literature have grouped the implementations on those where the MDP model is fully known and the ones where we have to infer this model by interaction with the environment (sampled-based implementations). The discussion on which cost function is better appears in several works [4], [5]. The most common approaches include the mean squared error (MSE), the mean squared Bellman error (MSBE) and the mean squared projected Bellman error (MSPBE). Not all of them are feasible in any scenario. MSE needs knowledge about the real value function and from there the linear approximation is straightforward through a projection, while MSBE and MSPBE make the linear approximation match the Bellman equation without any need of the value function. Different implementations of the optimization of these cost functions have been proposed in the literature (see, e.g., [4] and [6] and references therein) and their performance have been studied [7, 8]. Mixed linear MSBE and MSPBE strategies are also available in the literature [9], leading to of hybrid algorithms that may benefit from both criteria.

Setting the cost functions from where to start the value function approximation is the best way to provide a unified view for the different implementations in the literature. Precisely this unified view is one of the aims of this paper. The motivation for that is that if we are able to get a fixed point or closed form solution for the optimization of the cost function, further to provide a solution to that equation, we can also propose iterative aproaches based on that equation such as stochastic gradient descent or iterative solving of the fixed point equation.

Similarly to what was done in [5] with MSE and MSBE and their fixed point solutions named respectively Temporal Difference (TD) and Bellman residual (BR), here we provide a unified view of MSPBE and MSBE through a projection tool. Therefore our work would extend what was done in [5] by also analysing MSPBE and we will show that the optimization of both cost functions, MSBE and MSPBE, lead to a unique fixed point equation that just uses a parametrization of a oblique projection of the Bellman equation in the feature space. Furthermore it will be shown that proposing any iterative or sampled based method of this unified fixed point equation leads to many well known earlier method such as standard least-squares recursive approaches [1], and gradient based methods. In [10, 6] adaptive implementations for optimizing the MSPBE were proposed. Interestingly, it also noticed that a linear prediction of the conditioned expected Bellman error makes the instantaneous gradi-

[*]This work has been partly funded by the Spanish Ministry of Science and Innovation with the project GRE3N (TEC 2011-29006-C03-01/02/03) and in the program CONSOLIDER-INGENIO 2010 under project COMONSENS (CSD 2008-00010).

[†]This work was supported in part by the Spanish Ministry of Science and Innovation under the grants TEC2009-14219-C03-01,TEC2010-21217-C02-02-CR4HFDVL and in the program CONSOLIDER-INGENIO 2010 under the grant CSD2008-00010 COMONSENS; and by the European Commission under the grant FP7-ICT-2009-4-248894-WHERE-2.

ent of the MSBE identical to the instantaneous stochastic approximation of the gradient of the MSPBE. A few earlier works have highlighted this equivalence, but without entering in much detail (see e.g., [11]). Here we aim to clarify this relationship between the MSBE and the MSBPE and derive a new variation of the stochastic approximation algorithms introduced in [10] with improved performance.

2. VALUE FUNCTION APPROXIMATION

We consider the standard reinforcement learning framework where an agent learns by interaction with the environment. The environment is modelled by a MDP with a finite number of states $s \in \mathcal{S}$ and actions $a \in \mathcal{A}$. At time t, the transition probability from one state s to state s', when taking action a is given by $\mathcal{P}^a_{ss'} = \mathbb{P}\{s_{t+1} = s' | s_t = s, a_t = a\}$ and the reward obtained at this point is $\mathcal{R}^a_{ss'} = \mathbb{E}\{R_t | s_t = s, s_{t+1} = s', a_t = a\}$. We assume that the agent follows a policy π that determines his behaviour through the probability of taking action a when being at state s, $\pi(s,a) = \mathbb{P}\{a_t = a | s_t = s\}$, and we also assume that the state process is an irreducible and aperiodic Markov chain with stationary distribution induced by the policy π and given by the vector $\boldsymbol{\mu} = [\mu(1), \ldots, \mu(|\mathcal{S}|)]^\top$. Then the value function $V^\pi(s)$ is the expected accumulated reward that an agent would receive, when it starts from state s and follows policy π:

$$\begin{aligned} V^\pi(s) &= \mathbb{E}\left\{\sum_{k=0}^{\infty} \gamma^k R_{t+k} | s_t = s\right\} \\ &= \sum_a \pi(s,a) \sum_{s'} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma V^\pi(s')\right) \\ &= \sum_{a,s'} \pi(s,a) \mathcal{P}^a_{ss'} \mathcal{R}^a_{ss'} + \gamma \sum_{s'} \sum_a \pi(s,a) \mathcal{P}^a_{ss'} V^\pi(s') \\ &= r^\pi(s) + \gamma \sum_{s'} \mathcal{P}^\pi_{ss'} V^\pi(s') \end{aligned} \tag{1}$$

which can be expanded in vector form, as

$$\mathbf{V}^\pi = \mathbf{r}^\pi + \gamma \mathbf{P}^\pi \mathbf{V}^\pi = \mathcal{T}(\mathbf{V}^\pi) \tag{2}$$

This equation is the well-known Bellman equation and $\mathcal{T}(\cdot)$ is the so named Bellman operator [1]. From now on the dependence with policy π will be dropped for the sake of clarity.

2.1. Linear Value function approximation

The value function $V(s)$ can be linearly approximated with the help of parameter $\boldsymbol{\theta}$: $V_\theta(s) = \boldsymbol{\phi}^T(s)\boldsymbol{\theta}$, where the feature vector $\boldsymbol{\phi}(s) \in \mathbb{R}^{|\mathcal{F}|}$ is defined in a reduced space dimension $|\mathcal{F}| < |\mathcal{S}|$. The approximation subspace $\mathcal{S}_\Phi$ is the subspace spanned by $\boldsymbol{\Phi} = \left[\boldsymbol{\phi}^T(s_1) \ldots \boldsymbol{\phi}^T(s_{|\mathcal{S}|})\right]^T$: $\mathcal{S}_\Phi = \left\{\boldsymbol{\Phi}\mathbf{x} | \mathbf{x} \in \mathbb{R}^{|\mathcal{F}|}\right\}$. Hence, the value function approximation is given in vector form by

$$\mathbf{V}_\theta = \begin{bmatrix} \boldsymbol{\phi}^T(s_1) \\ \vdots \\ \boldsymbol{\phi}^T(s_{|\mathcal{S}|}) \end{bmatrix} \boldsymbol{\theta} = \boldsymbol{\Phi}\boldsymbol{\theta} \tag{3}$$

The optimal linear approximation to the value function $\mathbf{V}$, with respect to the weighted Euclidean norm $\|\cdot\|^2_{\boldsymbol{\Xi}}$, in the subspace $\mathcal{S}_\Phi$, is obtained from minimizing the MSE between the approximate value function and the true value function defined as:

$$\mathcal{J}_{\text{MSE}}(\boldsymbol{\theta}) = \|\mathbf{V} - \boldsymbol{\Phi}\boldsymbol{\theta}\|^2_{\boldsymbol{\Xi}} \tag{4}$$

where $\boldsymbol{\Xi} = \text{diag}(\boldsymbol{\mu})$ is a diagonal positive definite matrix and $\|\mathbf{x}\|^2_{\boldsymbol{\Xi}} = \mathbf{x}^T\boldsymbol{\Xi}\mathbf{x} = \sum_i \mu(i)x_i^2$. If matrix $\boldsymbol{\Phi}$ has linearly independent columns the solution is unique and is given by the projection of the value function with respect to the given weighted norm, denoted by $\Pi_{\boldsymbol{\Xi}} = \boldsymbol{\Phi}\left(\boldsymbol{\Phi}^T\boldsymbol{\Xi}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{\Xi}$, such that

$$\boldsymbol{\Phi}\boldsymbol{\theta} = \Pi_{\boldsymbol{\Xi}}\mathbf{V} \tag{5}$$

It should be noticed that the value function $\mathbf{V}$ will be rarely available for parameter estimation. Thus, alternative cost functions have to be considered, like the MSBE or the MSPBE.

2.1.1. Mean Squared Bellman Error

The solution to the Bellman equation in the subspace $\mathcal{S}_\Phi$ has been proposed in the literature as an indirect approach to obtain parameter $\boldsymbol{\theta}$. Thus we minimize the cost function defined as the mean squared Bellman error:

$$\begin{aligned} \mathcal{J}_{\text{MSBE}}(\boldsymbol{\theta}) &= \|\mathcal{T}(\boldsymbol{\Phi}\boldsymbol{\theta}) - \boldsymbol{\Phi}\boldsymbol{\theta}\|^2_{\boldsymbol{\Xi}} \\ &= (\mathcal{T}(\boldsymbol{\Phi}\boldsymbol{\theta}) - \boldsymbol{\Phi}\boldsymbol{\theta})^T \boldsymbol{\Xi} (\mathcal{T}(\boldsymbol{\Phi}\boldsymbol{\theta}) - \boldsymbol{\Phi}\boldsymbol{\theta}) \end{aligned} \tag{6}$$

In order to minimize (6) we obtain its gradient:

$$\nabla\mathcal{J}_{\text{MSBE}}(\boldsymbol{\theta}) = -\left((\mathbf{I} - \gamma\mathbf{P})\boldsymbol{\Phi}\right)^T \boldsymbol{\Xi} (\mathbf{r} + (\gamma\mathbf{P} - \mathbf{I})\boldsymbol{\Phi}\boldsymbol{\theta}) \tag{7}$$

From (7) we have:

$$\boldsymbol{\theta} = \left(\boldsymbol{\Phi}^T (\mathbf{I} - \gamma\mathbf{P})^T \boldsymbol{\Xi}\boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^T (\mathbf{I} - \gamma\mathbf{P})^T \boldsymbol{\Xi}\mathcal{T}(\boldsymbol{\Phi}\boldsymbol{\theta}) \tag{8}$$

Following a similar approach than in [5], a more compact formulation can be derived if we define the projection with a different weighted norm (i.e., an oblique projection):

$$\begin{aligned} \boldsymbol{\Phi}\boldsymbol{\theta} &= \boldsymbol{\Phi}\left(\boldsymbol{\Phi}^T (\mathbf{I} - \gamma\mathbf{P})^T \boldsymbol{\Xi}\boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^T (\mathbf{I} - \gamma\mathbf{P})^T \boldsymbol{\Xi}\mathcal{T}(\boldsymbol{\Phi}\boldsymbol{\theta}) \\ &= \Pi_{(\mathbf{I}-\gamma\mathbf{P})^T\boldsymbol{\Xi}}\mathcal{T}(\boldsymbol{\Phi}\boldsymbol{\theta}) \end{aligned} \tag{9}$$

Note that (9) allows for a fixed point equation representation of the optimal parameter solution. As it will be shown below, this is key for the unified treatment of the MSBE and MSPBE that we propose in this paper.

2.1.2. Mean Squared Projected Bellman Error

While the previous approach aims to find a vector lying in the subspace $\mathcal{S}_\Phi$ that satisfies the Bellman equation, another alternative is to first place the Bellman solution in the subspace by means of a projection and then to compute the closest vector by minimizing the so called mean squared projected Bellman error.

$$\begin{aligned} \mathcal{J}_{\text{MSPBE}}(\boldsymbol{\theta}) &= \|\Pi_{\boldsymbol{\Xi}}\mathcal{T}(\boldsymbol{\Phi}\boldsymbol{\theta}) - \boldsymbol{\Phi}\boldsymbol{\theta}\|^2_{\boldsymbol{\Xi}} \\ &= (\mathcal{T}(\boldsymbol{\Phi}\boldsymbol{\theta}) - \boldsymbol{\Phi}\boldsymbol{\theta})^T \Pi^T_{\boldsymbol{\Xi}}\boldsymbol{\Xi}\Pi_{\boldsymbol{\Xi}} (\mathcal{T}(\boldsymbol{\Phi}\boldsymbol{\theta}) - \boldsymbol{\Phi}\boldsymbol{\theta}) \end{aligned} \tag{10}$$

Given that $\Pi^T_{\boldsymbol{\Xi}}\boldsymbol{\Xi}\Pi_{\boldsymbol{\Xi}} = \left(\Pi^T_{\boldsymbol{\Xi}}\boldsymbol{\Xi}\Pi_{\boldsymbol{\Xi}}\right)^T = \boldsymbol{\Xi}\Pi_{\boldsymbol{\Xi}}$ we can optimize (10), obtaining:

$$\nabla\mathcal{J}_{\text{MSPBE}}(\boldsymbol{\theta}) = -\left((\mathbf{I} - \gamma\mathbf{P})\boldsymbol{\Phi}\right)^T \boldsymbol{\Xi}\Pi_{\boldsymbol{\Xi}} (\mathbf{r} + (\gamma\mathbf{P} - \mathbf{I})\boldsymbol{\Phi}\boldsymbol{\theta}) \tag{11}$$

From (11) we can obtain:

$$\begin{aligned}\boldsymbol{\theta} &= \left(\boldsymbol{\Phi}^T \left(\mathbf{I} - \gamma \mathbf{P}\right)^T \boldsymbol{\Xi} \Pi_{\boldsymbol{\Xi}} \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^T \left(\mathbf{I} - \gamma \mathbf{P}\right)^T \boldsymbol{\Xi} \Pi_{\boldsymbol{\Xi}} \mathcal{T}\left(\boldsymbol{\Phi}\boldsymbol{\theta}\right) \\ &= \left(\boldsymbol{\Phi}^T \left(\mathbf{I} - \gamma \Pi_{\boldsymbol{\Xi}} \mathbf{P}\right)^T \boldsymbol{\Xi} \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^T \left(\mathbf{I} - \gamma \Pi_{\boldsymbol{\Xi}} \mathbf{P}\right)^T \boldsymbol{\Xi} \mathcal{T}\left(\boldsymbol{\Phi}\boldsymbol{\theta}\right)\end{aligned} \tag{12}$$

And from (12) MSPBE fixed point solution can be written as:

$$\boldsymbol{\Phi}\boldsymbol{\theta} = \Pi_{\left(\mathbf{I} - \gamma \Pi_{\boldsymbol{\Xi}} \mathbf{P}\right)^T \boldsymbol{\Xi}} \mathcal{T}\left(\boldsymbol{\Phi}\boldsymbol{\theta}\right) \tag{13}$$

It should be noted the similarity of the MSBE solution, in (9), and the solution for the MSPBE, in (13), where the only difference is that in the first case we work directly with $\mathbf{P}$ and in the second case with the projected version of this matrix $\Pi_{\boldsymbol{\Xi}}\mathbf{P}$.

2.2. Weighted norm to model interaction with the environment

The weighted norm defined in (4) plays a fundamental role in the definition of average values of any of the parameters under study in the MDP. We have defined the elements within $\boldsymbol{\Xi}$ diagonal as the visitation probability $\mu\left(s\right)$ for each state s. Then we have the following equivalences [6]:

$$\mathbb{E}\left\{\boldsymbol{\phi}\boldsymbol{\phi}^T\right\} = \sum_s \mu\left(s\right) \boldsymbol{\phi}(s) \boldsymbol{\phi}(s)^T = \boldsymbol{\Phi}^T \boldsymbol{\Xi} \boldsymbol{\Phi} \tag{14}$$

$$\begin{aligned}\mathbb{E}\left\{\boldsymbol{\phi}'\boldsymbol{\phi}^T\right\} &= \sum_{s,s'} \mu(s) \mathcal{P}_{ss'} \boldsymbol{\phi}(s') \boldsymbol{\phi}(s)^T \\ &= \sum_s \mu(s) \boldsymbol{\phi}'(s) \boldsymbol{\phi}(s)^T \\ &= \left(\mathbf{P}\boldsymbol{\Phi}\right)^T \boldsymbol{\Xi} \boldsymbol{\Phi} = \boldsymbol{\Phi}'^T \boldsymbol{\Xi} \boldsymbol{\Phi}\end{aligned} \tag{15}$$

$$\begin{aligned}\mathbb{E}\left\{e\left(\boldsymbol{\theta}\right)\boldsymbol{\phi}\right\} &= \sum_s \mu(s) \boldsymbol{\phi}(s) \\ &\quad \left(V_\theta(s) - r(s) - \gamma \sum_{s'} \mathcal{P}_{ss'} V_\theta(s')\right) \\ &= \boldsymbol{\Phi}^T \boldsymbol{\Xi} \left(\boldsymbol{\Phi}\boldsymbol{\theta} - \mathcal{T}\left(\boldsymbol{\Phi}\boldsymbol{\theta}\right)\right)\end{aligned} \tag{16}$$

$$\begin{aligned}\mathbb{E}\left\{e\left(\boldsymbol{\theta}\right)\boldsymbol{\phi}'\right\} &= \sum_{s,s'} \mu(s) \mathcal{P}_{ss'} \boldsymbol{\phi}(s) \\ &\quad \left(V_\theta(s) - r(s) - \gamma \sum_{s''} \mathcal{P}_{ss''} V_\theta(s'')\right) \\ &= \left(\mathbf{P}\boldsymbol{\Phi}\right)^T \boldsymbol{\Xi} \left(\boldsymbol{\Phi}\boldsymbol{\theta} - \mathcal{T}\left(\boldsymbol{\Phi}\boldsymbol{\theta}\right)\right) \\ &= \boldsymbol{\Phi}'^T \boldsymbol{\Xi} \left(\boldsymbol{\Phi}\boldsymbol{\theta} - \mathcal{T}\left(\boldsymbol{\Phi}\boldsymbol{\theta}\right)\right)\end{aligned} \tag{17}$$

where $e\left(\boldsymbol{\theta}\right)$ is a random variable that models the error between the value function approximation and the Bellman equation; $\boldsymbol{\phi}$ and $\boldsymbol{\phi}'$ are random variables that take values in the feature vector space $\boldsymbol{\phi}(s)$ with $s \in \mathcal{S}$. $\boldsymbol{\phi}$ represents the "present" feature while $\boldsymbol{\phi}'$ represents the "future" feature. Given that in the Bellman equation in (2), the matrix $\mathbf{P}$ helps to obtain the expected accumulated future reward, once the immediate reward $\mathbf{r}$ has been obtained, similarly, in the feature space the matrix equivalence obtained in (15) can be interpreted as the feature space $\boldsymbol{\Phi}' = \mathbf{P}\boldsymbol{\Phi}$ after transition to the future states.

We will use the equivalence between the expected values (14)-(17) and their matrix forms to derive sample-based stochastic approximations of the fixed point equations and gradients introduced in Section 2, as well as to highlight some degree of equivalence between the MSBE and the MSPBE.

2.3. MSBE and MSPBE equivalence under linear prediction of features

When the MDP model is fully known, there would be no need for an estimation of the future features given that $\boldsymbol{\Phi}' = \mathbf{P}\boldsymbol{\Phi}$ would be fully known. However, in many cases [11] it has been proposed to use the present feature space to make a linear prediction of the future features $\boldsymbol{\phi}' \approx \mathscr{P}_{\boldsymbol{\Phi}}\boldsymbol{\phi}$. This fact is of relevance when it is applied to sampled-based implementations [6] where the environment model is learnt by means of agent interactions with the environment.

If the linear predictor is defined to minimize the MSE between $\boldsymbol{\phi}'$ and $\mathscr{P}_{\boldsymbol{\Phi}}\boldsymbol{\phi}$, then:

$$\mathscr{P}_{\boldsymbol{\Phi}} = \mathbb{E}\left\{\boldsymbol{\phi}'\boldsymbol{\phi}^T\right\} \mathbb{E}\left\{\boldsymbol{\phi}\boldsymbol{\phi}^T\right\}^{-1} = \left(\mathbf{P}\boldsymbol{\Phi}\right)^T \boldsymbol{\Xi} \boldsymbol{\Phi} \left(\boldsymbol{\Phi}^T \boldsymbol{\Xi} \boldsymbol{\Phi}\right)^{-1} \tag{18}$$

In the feature space we would have the following approximation:

$$\boldsymbol{\Phi}' = \mathbf{P}\boldsymbol{\Phi} \approx \boldsymbol{\Phi}\mathscr{P}_{\boldsymbol{\Phi}}^T = \boldsymbol{\Phi} \left(\boldsymbol{\Phi}^T \boldsymbol{\Xi} \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^T \boldsymbol{\Xi} \boldsymbol{P} \boldsymbol{\Phi} = \Pi_{\Xi} \mathbf{P} \boldsymbol{\Phi} \tag{19}$$

If we apply this equivalence to the fixed point equation (9), derived from the MSBE, it can be easily shown that projection $\Pi_{\left(\mathbf{I} - \gamma \mathbf{P}\right)^T \boldsymbol{\Xi}}$ turns into the projection $\Pi_{\left(\mathbf{I} - \gamma \Pi_{\boldsymbol{\Xi}} \mathbf{P}\right)^T \boldsymbol{\Xi}}$ which was used for the fixed point equation representation of the MSPBE given by (13). It should be noted though, that this equivalence between MSBE and MSPBE is only valid in those scenarios where it makes sense to apply the linear prediction, i.e. sample-based implementations of MSBE an MSPBE solutions.

3. ADAPTIVE IMPLEMENTATIONS

In this section we derive unified solutions for minimizing the MSBE and the MSPBE introduced in Section 2 in an *adaptive* manner: namely, a gradient-descent like algorithm and a iterative fixed-point equation. It should be noted that still the iterative methods do not suffice to get the value function approximation in those cases where the model of the environment is not fully avalable, thus the agents are able to learn directly from the stream of samples. We will show along this section, that both iterative approaches can be implemented following the general scheme in Figure 1.

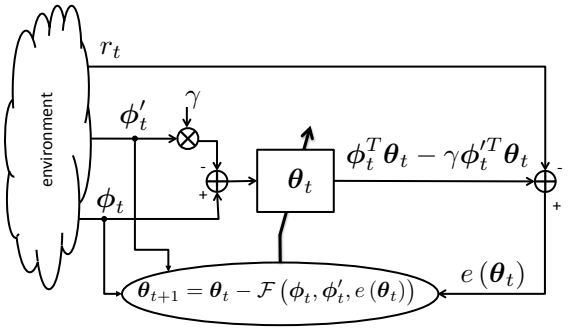


Fig. 1. General adaptive scheme for value function estimation.

3.1. Gradient-based iterative implementation

Vector parameter estimation through gradient descent on a arbitrary cost function can be defined:

$$\begin{aligned}\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t &= \alpha_t \nabla \mathcal{J}(\boldsymbol{\theta}_t) = \alpha_t \boldsymbol{\Phi}^T \mathbf{B}^T \mathbf{D}(\boldsymbol{\Phi}\boldsymbol{\theta}_t - \mathcal{T}(\boldsymbol{\Phi}\boldsymbol{\theta}_t)) \\ &= \alpha_t \boldsymbol{\Phi}^T \mathbf{B}^T \mathbf{D}\mathbf{e}(\boldsymbol{\theta}_t)\end{aligned} \tag{20}$$

where we have defined the error $\mathbf{e}(\boldsymbol{\theta}_t)$ between the value function approximation and the Bellman equation, α_t is the step-size for the gradient descent and:

- $\mathbf{B} = (\mathbf{I} - \gamma\mathbf{P})$ and $\mathbf{D} = \boldsymbol{\Xi}$ in the MSBE solution (see (7)).
- $\mathbf{B} = (\mathbf{I} - \gamma\mathbf{P})$ and $\mathbf{D} = \boldsymbol{\Xi}\boldsymbol{\Pi}_{\boldsymbol{\Xi}}$ in the MSPBE solution (see (11)).

Following the equivalences in (14)-(17), the gradients can be rewritten as follows:

$$\nabla \mathcal{J}_{\text{MSBE}}(\boldsymbol{\theta}) = \mathbb{E}\{e(\boldsymbol{\theta})\boldsymbol{\phi}\} - \gamma \mathbb{E}\{e(\boldsymbol{\theta})\boldsymbol{\phi}'\} \tag{21}$$

$$\nabla \mathcal{J}_{\text{MSPBE}}(\boldsymbol{\theta}) = \mathbb{E}\{e(\boldsymbol{\theta})\boldsymbol{\phi}\} - \gamma \mathbb{E}\left\{\boldsymbol{\phi}'\boldsymbol{\phi}^T\right\} \mathbb{E}\left\{\boldsymbol{\phi}\boldsymbol{\phi}^T\right\}^{-1} \mathbb{E}\{e(\boldsymbol{\theta})\boldsymbol{\phi}\} \tag{22}$$

If we used the linear predictor defined in (19) we would have that $\nabla \mathcal{J}_{\text{MSPBE}}(\boldsymbol{\theta}) = \nabla \mathcal{J}_{\text{MSBE}}(\boldsymbol{\theta})$ and therefore a unified view of both approaches. Thus for sample-based implementations we will focus on MSPBE implementations by means of (22).

3.1.1. LPBR implementation

By interaction with the environment, at time step t the agent receives data samples in the form of the triplets $(\boldsymbol{\phi}_t, r_t, \boldsymbol{\phi}'_t)$, where $\boldsymbol{\phi}_t$ and $\boldsymbol{\phi}'_t$ are the feature associated to state s_t and s_{t+1} respectively, r_t is the immediate value reward obtained and so we can compute $e_t(\boldsymbol{\theta}_t) = \boldsymbol{\phi}_t^T\boldsymbol{\theta}_t - (r_t + \gamma\boldsymbol{\phi}_t'^T\boldsymbol{\theta}_t)$.

The agent can use these samples to estimate the expected values in (22) differently. Our proposed algorithm for the iteration is to upgrade the mean estimates at each time sample as follows:

$$\mathbb{E}\left\{\boldsymbol{\phi}\boldsymbol{\phi}^T\right\} \approx \widehat{R}_{\boldsymbol{\Phi},t} = \frac{1}{t+1}\sum_{k=0}^{t} \boldsymbol{\phi}_k\boldsymbol{\phi}_k^T \tag{23}$$

$$\mathbb{E}\left\{\boldsymbol{\phi}'\boldsymbol{\phi}^T\right\} \approx \widehat{R}_{\boldsymbol{\Phi}'\boldsymbol{\Phi},t} = \frac{1}{t+1}\sum_{k=0}^{t} \boldsymbol{\phi}'_k\boldsymbol{\phi}_k^T \tag{24}$$

$$\mathbb{E}\{e(\boldsymbol{\theta})\boldsymbol{\phi}\} \approx \mathbf{e}_{\boldsymbol{\Phi},t} = \frac{1}{t+1}\sum_{k=0}^{t} e_k(\boldsymbol{\theta}_k)\boldsymbol{\phi}_k \tag{25}$$

which should be implemented in an efficient recursive manner. And finally define the iterative linear prediction Bellman residual (LPBR) algorithm:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t\left(\mathbf{e}_{\boldsymbol{\Phi},t} - \gamma\widehat{R}_{\boldsymbol{\Phi}'\boldsymbol{\Phi},t}\widehat{R}_{\boldsymbol{\Phi},t}^{-1}\mathbf{e}_{\boldsymbol{\Phi},t}\right) \tag{26}$$

3.1.2. TDC implementation [6]

If we approximate all the averages in (22) by its instantaneous values we obtain the so called TDC algorithm in [6]:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t\left(e_t\boldsymbol{\phi}_t - \gamma\boldsymbol{\phi}'_t\boldsymbol{\phi}_t^T w_t\right) \tag{27}$$

in which w_t is a long term estimate computed in a slower time scale as:

$$w_t = w_{t-1} + \beta_t\left(e_{t-1} - \boldsymbol{\phi}_{t-1}^T w_{t-1}\right)\boldsymbol{\phi}_{t-1} \tag{28}$$

3.2. Fixed-point Iterative implementation

Closed form formulation in terms of the projection of the Bellman equation such as those obtained in (5), (9) or (13) would allow a fixed point iterative implementation. Focusing on the MSE approach in (5), we have a least squares solution as the one proposed in [1]:

$$\boldsymbol{\theta}_{t+1} = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^{|\mathcal{F}|}} \|\mathcal{T}(\boldsymbol{\Phi}\boldsymbol{\theta}_t) - \boldsymbol{\Phi}\boldsymbol{\theta}\|_{\boldsymbol{\Xi}}^2 \tag{29}$$

The iterative solution:

$$\boldsymbol{\Phi}\boldsymbol{\theta}_{t+1} = \boldsymbol{\Pi}_{\boldsymbol{\Xi}}\mathcal{T}(\boldsymbol{\Phi}\boldsymbol{\theta}_t) \tag{30}$$

that can be formulated as follows:

$$\begin{aligned}\boldsymbol{\theta}_{t+1} &= \left(\boldsymbol{\Phi}^T\boldsymbol{\Xi}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{\Xi}(\mathbf{r} + \gamma\mathbf{P}\boldsymbol{\Phi}\boldsymbol{\theta}_t) \\ &= \boldsymbol{\theta}_t - \left(\boldsymbol{\Phi}^T\boldsymbol{\Xi}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{\Xi}(\boldsymbol{\Phi}\boldsymbol{\theta}_t - (\mathbf{r} + \gamma\mathbf{P}\boldsymbol{\Phi}\boldsymbol{\theta}_t)) \\ &= \boldsymbol{\theta}_t - \left(\boldsymbol{\Phi}^T\boldsymbol{\Xi}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{\Xi}\mathbf{e}(\boldsymbol{\theta}_t)\end{aligned} \tag{31}$$

3.2.1. Sample-based fixed point iterative implementation

Similarly to what was done in section 3.1, the correction term in (31) can be rewritten as follows:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \mathbb{E}\left\{\boldsymbol{\phi}\boldsymbol{\phi}^T\right\}^{-1}\mathbb{E}\{e(\boldsymbol{\theta})\boldsymbol{\phi}\} \tag{32}$$

The terms $\mathbb{E}\left\{\boldsymbol{\phi}\boldsymbol{\phi}^T\right\}^{-1}$ and $\mathbb{E}\{e(\boldsymbol{\theta})\boldsymbol{\phi}\}$ could be estimated from the triplet $(\boldsymbol{\phi}_t, r_t, \boldsymbol{\phi}'_t)$. Different approaches are available in the literature, one of them is the RLSTD implementation [1, 12].

4. SIMULATIONS

We study the performance of the proposed LPBR algorithm by simulation in a classical problem and compare it with other proposals in the literature such as RLSTD algorithm [12] or TDC algorithm [6].

Our MDP is a Markov chain of 7 states [2], with initial state in the middle of the chain (s_3), and with the two ends (s_0 and s_6) being terminal, absorbing states. There are only two possible actions, going *left* or *right*, which make the agent transit to the previous or next state in the chain, respectively (see Figure 2). Our goal is to predict the approximated state-value function for an uniform target policy (i.e., at every state the agent can choose left or right with equal probability). The figure of merit is the MSPBE.

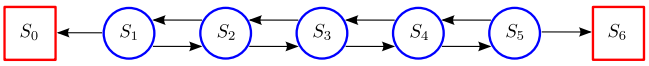
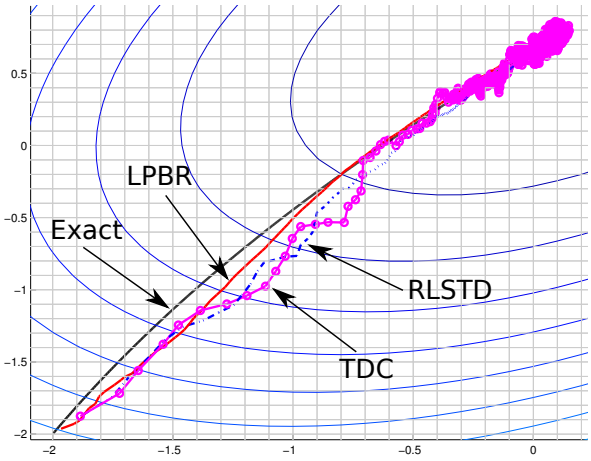


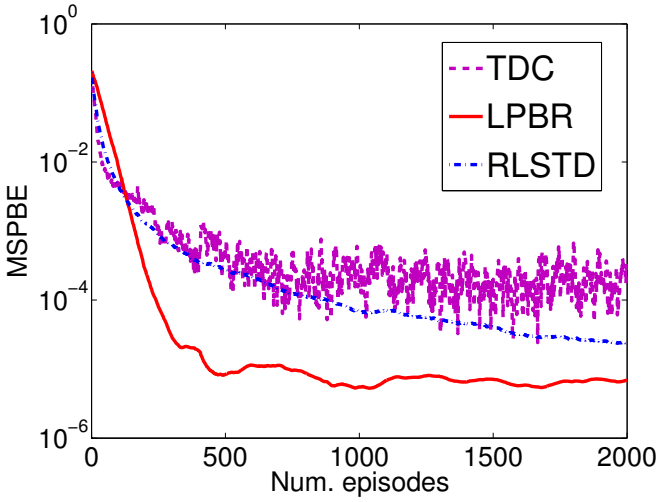
Fig. 2. State diagram of the random walk problem.

For the simulations the agent always advances in the direction it has moved, so the transition probabilities are $\mathbb{P}(s_{i+1}|s_i, \text{right}) = 1$, $\mathbb{P}(s_{i-1}|s_i, \text{left}) = 1$ and zero for any other case, except for the absorbing states for which $\mathbb{P}(s_0|s_0) = \mathbb{P}(s_6|s_6) = 1$. We choose a set of 2-dimensional handcrafted features to represent the state, which are $\boldsymbol{\phi}(s_0) = [0,0]^\top$, $\boldsymbol{\phi}(s_1) = [1,0]^\top$, $\boldsymbol{\phi}(s_2) = [\frac{1}{2},0]^\top$, $\boldsymbol{\phi}(s_3) = [\frac{1}{3},\frac{1}{3}]^\top$, $\boldsymbol{\phi}(s_4) = [0,\frac{1}{2}]^\top$, $\boldsymbol{\phi}(s_5) = [0,1]^\top$, and $\boldsymbol{\phi}(s_6) = [0,0]^\top$. Step-sizes for gradient descent is constant $\alpha_t = 0.1$ for all the algorithms and $\beta_t = 0.01$ in (28). The discount factor is $\gamma = 1$.

Some performance results are given in Figures 3.a and 3.b where we see that the proposed LPBR is very competitive, even with respect to RLSTD which has similar complexity. We also appreciate that TDC shows more variance and bias than LPBR and RLSTD. This is natural as, though TDC approximates a long-term estimate of two of the expected values in (28), it still approximates the other statistics in (22) instantaneously. RLSTD is more accurate than TDC, and the proposed LPBR is even better. Note that, though the per-time complexity is $\mathcal{O}(|\mathcal{F}|)$ in these algorithms (where $|\mathcal{F}|$ is the dimension of the features), the less bias and variance of RLSTD and LPBR comes at the cost of more memory requirements, $\mathcal{O}(|\mathcal{F}|^2)$, which contrasts with the linear memory requirements of of TDC.



(a) Evolution of the gradient on a MSPBE surface.



(b) Error curves

Fig. 3. Random walk in a Markov chain. (a) Evolution of the exact (20) and stochastic approximations (Algorithm 6 in [12], (27) and (26)) of the gradient, and (b) Error curves.

5. CONCLUSIONS AND FUTURE WORK

We have presented a fixed point solution for the two typical cost functions for linear value prediction in the literature, providing a projection tool that shows the equivalence of the MSPBE and the MSBE with linear prediction of future features. From this analysis, we derived an efficient adaptive implementation that provides an unbiased linear estimate, showing competitive results through simulation in a classical domain.

This same approach could be extended in other directions, such as the multi-step TD(λ) family of algorithms [2], off-policy iteration and even distributed adaptive implementations. Moreover, so far we have only considered the policy evaluation problem, in which an agent predicts the goodness of a certain policy, a natural extension of our work would be to extend the same methodology to the control problem, using features that represent the state-action pairs and including a policy-update step.

6. REFERENCES

[1] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 2, ch. 6 -updated online- of *Athena Scientific Optimization and Computation Series*, Athena Scientific, 2005.

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Adaptive computation and machine learning. MIT Press, 1998.

[3] E. Altman, "Applications of Markov Decision Processes in Communications Networks: a Survey," Tech. Rep., Institut National de Recherche en Informatique et en Automatique (INRIA), 2000.

[4] M. Geist and O. Pietquin, "Parametric Value Function Approximation: a Unified View," in *Proceedings of the IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, April 2011.

[5] B. Scherrer, "Should one compute the temporal difference fix point or minimize the Bellman residual? The unified oblique projection view," in *International Conference on Machine Learning*, June 2010.

[6] H. R. Maei, *Gradient Temporal-Difference Learning Algorithms*, Ph.D. thesis, University of Alberta, 2011.

[7] H. Yu and D.P. Bertsekas, "Error bounds for approximations from projected linear equations," *Mathematics of Operations Research*, vol. 35, pp. 306–329, 2010.

[8] S. Mannor, D. Simester, P. Sun, and J. N. Tsitsiklis, "Bias and Variance Approximation in Value Function Estimates," *Management Science*, vol. 53, no. 2, pp. 308–322, February 2007.

[9] J. Johns, M. Petrik, and S. Mahadevan, "Hybrid least-Squares Algorithms for approximate Policy Evaluation," *Machine Learning*, vol. 76, pp. 243–256, 2009.

[10] R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvari, and E. Wiewiora, "Fast gradient-descent methods for temporal-difference learning with linear function approximation," in Proc. *International Conference on Machine Learning*, Montreal, Quebec, Canada, 2009, pp. 993–1000.

[11] R. Parr, L. Li, G. Taylor, C. Painter-Wakefield, and M. L. Littman, "An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning," in *Proceedings of the Twenty-Fifth International Conference*, 2008, pp. 752–759.

[12] C. Szepesvari, *Algorithms for Reinforcement Learning*, Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2009.