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 of certain sequence of actions (policy) using linear function approximation of the value function. Under the paradigm of Stochastic control and reinforcement learning, our analysis shows the equivalence of the mean squared projected Bellman error and the mean squared Bellman error with linear prediction of future features. Indeed, this analysis induces an efficient adaptive implementation that provides fast and 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— Mean squared Bellman Error, Mean squared projected Bellman Error, Linear value function.

1. INTRODUCCTION

The number of applications of Markov Decision Processes (MDP) in many fields is overwhelming. Particularly in communications and signal processing, most of the problems where the enviroment could be represented as a set of states with some transition probabilities among them and some associated rewards to represent how much desirable for the agent is to be in such state, can be formulated as a generic MDP. Sequential decision / estimation / tracking problems, active monitoring in a wireless sensor networks to keep a certain parameter in a certain range where some actions are applicable, along with networking issues as call admission control, packet admission control, congestion control, routing, scheduling of services could be selected [1]. Also, the formulation in terms of features instead of states allows the applications of these techniques to a much larger number of applications even with continuous state space or very large discrete number of states. Indeed, the features may be able to represent the measurements of a certain problem to represent the information about the environment even if the underlying state is not observable (like in partially observable MDP).

The fundamentals of the topic addressed on this paper are clearly related to the Dynamic Programming, Optimal Control and Reinforcement learning disciplines [2][3]. In those text books, the need of linear approximation of value (and also learning) functions is clearly stated and a subsequent collection of papers have been focused on this topic. In particular, several important discussions have been described related to two main issues: the selection of the cost function to be optimized and how to approximate these functions using samples guaranteeing fast and unbiased implementations. These two discussions are for instance addressed in [4] where known algorithms are classified as belonging to i) boostrapping approach, ii) stochastic gradient descent and recursive least squares, iii) residual approaches also including stochastic gradient and recursive least squares implementation and iv) projection fixed-point approaches with several iterative implementations. Although very interesting as an overview, a deep discussion about the selection of the cost function to be optimized is not properly addressed.

On the other hand, reference [5] presented a unified oblique projection view where the mean squared error (MSE), the mean squared Bellman error (MSBE) and the mean squared projected Bellman error (MSPBE) are particular cases of some oblique projections of the exact value function on the space spanned by the features. Reference [6] considers the possibility of optimizing a linear combination of the MSBE and MSPBE, leading to a kind of hybrid algorithms that may benefit from both criteria. In [7, 8] some performance bounds are provided, which suggest the superiority of the MSPBE over other criteria.

Reference [9] proposes adaptive implementations for optimizing different criteria, including the MSBE and the MSPBE. Interestingly, it also noticed that a linear prediction of the conditioned expected Bellman error makes the instantaneous gradient of the MSBE identical to the instantaneous stochastic approximation of the gradient of the MSPBE, introducing the temporal-difference-with-correction (TDC) algorithm, that we will compare with in Section 4. This equivalence is the starting point of this paper. Here we aim to clarify this relationship so we can derive a new variation of the instantaneous approximations introduced in [9] with improved performance. A few earlier works have highlighted this equivalence, but without entering in much detail (see e.g., [10]).

Our main contributions is a unified analysis that connects the MSBE and the MSPBE, in the same line as [4]-[6], [10], with rigorous treatment. A fixed point solution is given for both cases using weighted projections and from this common projection tool it is easily proved that both value functions are equivalent given that the features of the future state, after state transition, can be approximated as a linear model of the features of the current state. The fixed

[*] 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.

point solutions is further linked with standard least-squares recursive approaches [2], and with gradient based methods. The solution of both approaches should be the same, and indeed is the temporal-difference solution which has been already analyzed in many works (see e.g., [7, 8].

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 $a \in \mathcal{A}$. In time instant t, the transition probability from one state s to state s', when taking action a is given by $\mathcal{P}^a_{ss'} = \mathbb{P}\left\{s_{t+1} = s' | s_t = s, a_t = a\right\}$ and the reward obtained at this point is $\mathcal{R}^a_{ss'} = \mathbb{E}\left\{R_t | s_t = s, s_{t+1} = s', a_t = a\right\}$. If the agent follows a policy π that determines his behaviour through the probability of taking action a when being at state s, $\pi(s,a) = \mathbb{P}\left\{a_t = a | s_t = s\right\}$, 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

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

where $\mathcal{T}(\cdot)$ is the well known Bellman operator. 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 $|\mathcal{F}| < |\mathcal{S}|$. 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:

$$\|\mathbf{V} - \boldsymbol{\Phi}\boldsymbol{\theta}\|^2_{\boldsymbol{\Xi}} \tag{4}$$

where $\|\mathbf{x}\|^2_{\boldsymbol{\Xi}} = \mathbf{x}^T\boldsymbol{\Xi}\mathbf{x}$ and $\boldsymbol{\Xi}$ is a diagonal positive definite matrix. 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, $\mathcal{J}(\boldsymbol{\theta})$, have to be considered, like 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:

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

Minimizing (6) we obtain:

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

Note that (7) allows for a fixed point equation representation of the optimal parameter solution, which will be key for the posterior analysis of the solutions to the different cost functions, and to show equivalences between them. From (7) we have:

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

A more compact formulation can be derived if we define the projection with a different weighted norm:

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

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.

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

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 solve (10), obtaining:

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

From (11) we can obtain:

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

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

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

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. Linear prediction of future features: MSPBE and MSBE equivalence

In the Bellman equation given by (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 $\mathbf{P}\boldsymbol{\Phi}$ could be interpreted as the feature space $\boldsymbol{\Phi}'$ after transition to the future states. When the MDP model is fully known, there would be no need for an estimation of the future features, however, in many cases it has been proposed to use the feature space to make a linear prediction of the future features [10]. This fact is of relevance when it is applied to sampled-based implementations [9] where the environment model is learnt by means of agent interactions with the environment. The linear prediction $\mathbf{P}_{\boldsymbol{\Phi}}$ is given by:

$$\boldsymbol{P}\boldsymbol{\Phi} \approx \boldsymbol{\Phi}\mathbf{P}_{\boldsymbol{\Phi}} = \boldsymbol{\Phi}\left(\boldsymbol{\Phi}^T\boldsymbol{\Xi}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{\Xi}\boldsymbol{P}\boldsymbol{\Phi} = \Pi_{\Xi}\boldsymbol{P}\boldsymbol{\Phi} \tag{14}$$

Applying this prediction to the fixed point MSBE equation in (9) it can be easily shown that projection $\Pi_{(\mathbf{I}-\gamma\mathbf{P})^T\boldsymbol{\Xi}}$ turns into the projection used for MSPBE fixed point equation $\Pi_{(\mathbf{I}-\gamma\Pi_{\boldsymbol{\Xi}}\mathbf{P})^T\boldsymbol{\Xi}}$ and consequently both parameter value estimations would be equivalent.

3. ITERATIVE AND SAMPLED-BASED IMPLEMENTATIONS

Each of the previous cost function provides a different solutions for the optimal parameter vector. Iterative methods applied to solving the parameter vector arise in different scenarios such as those cases where the number of states $|\mathcal{S}|$ is large or where we do not have full access to the model of the environment. Closed form formulation in terms of the projection of the Bellman equation such as those in (5), (9) or (13) lead to a least squares solution as the one proposed in [2]:

$$\boldsymbol{\theta}_{t+1} = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^{\mathcal{F}|}} \|\mathbf{A}\mathcal{T}(\boldsymbol{\Phi}\boldsymbol{\theta}_t) - \boldsymbol{\Phi}\boldsymbol{\theta}\|_{\boldsymbol{\Xi}}^2 \tag{15}$$

where $\mathbf{A} = \mathbf{I}$ or $\mathbf{A} = \Pi_{\boldsymbol{\Xi}}$ for a MSBE-like or MSPBE-like solution respectively. Both implementations lead to the same iterative solution:

$$\boldsymbol{\Phi}\boldsymbol{\theta}_{t+1} = \Pi_{\boldsymbol{\Xi}}\mathbf{A}\mathcal{T}(\boldsymbol{\Phi}\boldsymbol{\theta}_t) = \Pi_{\boldsymbol{\Xi}}\mathcal{T}(\boldsymbol{\Phi}\boldsymbol{\theta}_t) \tag{16}$$

where

$$\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t = \left(\boldsymbol{\Phi}^T\boldsymbol{\Xi}\boldsymbol{\Phi}\right)^{-1}(\mathbf{d} - \mathbf{C}\boldsymbol{\theta}_t) \tag{17}$$

with $\mathbf{C} = \left(\boldsymbol{\Phi}^T\boldsymbol{\Xi}\left(\mathbf{I} - \gamma\mathbf{P}^{\pi}\right)\boldsymbol{\Phi}\right)$ and $\mathbf{d} = \boldsymbol{\Phi}^T\boldsymbol{\Xi}\mathbf{r}^{\pi}$.

Another iterative solution for vector parameter are gradient based solutions where:

$$\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t = \alpha_t\nabla\mathcal{J}(\boldsymbol{\theta}_t) = \alpha_t\boldsymbol{\Phi}^T\mathbf{B}^T\mathbf{D}\left(\boldsymbol{\Phi}\boldsymbol{\theta}_t - \mathcal{T}(\boldsymbol{\Phi}\boldsymbol{\theta}_t)\right) \tag{18}$$

where:

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

3.1. Sampled-based value function iteration

The parameter value iteration given in (18) still needs full knowledge of the dynamic model of the environment. However, if we assume that the state visitation probability $\mu(s)$ when interacting with the environment is known, we can choose $\boldsymbol{\Xi} = \operatorname{diag}\left(\mu(s_1), \mu(s_2), \mu\left(s_{|\mathcal{S}|}\right)\right)$ and we have the following equivalences [9]:

$$\mathbb{E}\left\{\boldsymbol{\phi}\boldsymbol{\phi}^T\right\} = \sum_s \mu(s)\,\boldsymbol{\phi}(s)\,\boldsymbol{\phi}(s)^T = \boldsymbol{\Phi}\boldsymbol{\Xi}\boldsymbol{\Phi}^T \tag{19}$$

$$\begin{aligned}\mathbb{E}\left\{\boldsymbol{\phi}'\boldsymbol{\phi}^T\right\} &= \sum_{s,s'} \mu(s)\,\mathcal{P}_{ss'}\boldsymbol{\phi}(s')\,\boldsymbol{\phi}(s)^T \\ &= \sum_s \mu(s)\,\boldsymbol{\phi}'(s)\,\boldsymbol{\phi}(s)^T \\ &= (\mathbf{P}\boldsymbol{\Phi})^T\boldsymbol{\Xi}\boldsymbol{\Phi} = \boldsymbol{\Phi}'^T\boldsymbol{\Xi}\boldsymbol{\Phi}\end{aligned} \tag{20}$$

$$\begin{aligned}\mathbb{E}\{e(\boldsymbol{\theta})\,\boldsymbol{\phi}\} &= \sum_s \mu(s)\,\boldsymbol{\phi}(s)\left(r(s) + \gamma\sum_{s'}\mathcal{P}_{ss'}V_\theta(s') - V_\theta(s)\right) \\ &= \boldsymbol{\Phi}^T\boldsymbol{\Xi}\left(\boldsymbol{\Phi}\boldsymbol{\theta} - \mathcal{T}(\boldsymbol{\Phi}\boldsymbol{\theta})\right)\end{aligned} \tag{21}$$

$$\begin{aligned}\mathbb{E}\{e(\boldsymbol{\theta})\,\boldsymbol{\phi}'\} &= \sum_{s,s'} \mu(s)\,\mathcal{P}_{ss'}\boldsymbol{\phi}(s)\left(r(s) + \gamma\sum_{s''}\mathcal{P}_{ss''}V_\theta(s'') - V_\theta(s)\right) \\ &= (\mathbf{P}\boldsymbol{\Phi})^T\boldsymbol{\Xi}\left(\boldsymbol{\Phi}\boldsymbol{\theta} - \mathcal{T}(\boldsymbol{\Phi}\boldsymbol{\theta})\right) \\ &= \boldsymbol{\Phi}'^T\boldsymbol{\Xi}\left(\boldsymbol{\Phi}\boldsymbol{\theta} - \mathcal{T}(\boldsymbol{\Phi}\boldsymbol{\theta})\right)\end{aligned} \tag{22}$$

Thus the gradients can be rewritten as follows:

$$\nabla\mathcal{J}_{\text{MSBE}}(\boldsymbol{\theta}) = \mathbb{E}\{e(\boldsymbol{\theta})\,\boldsymbol{\phi}\} - \gamma\mathbb{E}\{e(\boldsymbol{\theta})\,\boldsymbol{\phi}'\} \tag{23}$$

$$\nabla\mathcal{J}_{\text{MSPBE}}(\boldsymbol{\theta}) = \mathbb{E}\{e(\boldsymbol{\theta})\,\boldsymbol{\phi}\} - \gamma\mathbb{E}\left\{\boldsymbol{\phi}'\boldsymbol{\phi}^T\right\}\mathbb{E}\left\{\boldsymbol{\phi}\boldsymbol{\phi}^T\right\}^{-1}\mathbb{E}\{e(\boldsymbol{\theta})\,\boldsymbol{\phi}\} \tag{24}$$

Using (14) we get the equivalence $\nabla\mathcal{J}_{\text{MSPBE}}(\boldsymbol{\theta}) = \nabla\mathcal{J}_{\text{MSBE}}(\boldsymbol{\theta})$.

By interaction with the environment, at time step t we have access to the triplet $(\boldsymbol{\phi}_t, r_t, \boldsymbol{\phi}'_t)$, 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 $e_t = r_t + \gamma\boldsymbol{\phi}'^T_t\boldsymbol{\theta}_t - \boldsymbol{\phi}^T_t\boldsymbol{\theta}_t$. From these values we can estimate the expected vlaues in (19)-(22) leading to different sampled-based algorithms.

If we approximate all the averages in (24) by its instantaneous values we obtain the so called TDC algorithm in [9]:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t\left(e_t\boldsymbol{\phi}_t - \gamma\boldsymbol{\phi}'_t\boldsymbol{\phi}^T_t w_t\right) \tag{25}$$

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

$$w_t = w_{t-1} + \beta_t\left(e_{t-1} - \boldsymbol{\phi}^T_{t-1}w_{t-1}\right)\boldsymbol{\phi}_{t-1} \tag{26}$$

Our proposal for the iteration is to upgrade the mean estimates at each time sample as follows:

$$\mathbb{E}\left\{\boldsymbol{\phi}\boldsymbol{\phi}^T\right\} \approx \widehat{R}_{\boldsymbol{\Phi},t} = \frac{1}{t+1}\sum_{k=0}^{t}\boldsymbol{\phi}_k\boldsymbol{\phi}^T_k \tag{27}$$

$$\mathbb{E}\left\{\boldsymbol{\phi}'\boldsymbol{\phi}^T\right\} \approx \widehat{R}_{\boldsymbol{\Phi}'\boldsymbol{\Phi},t} = \frac{1}{t+1}\sum_{k=0}^{t}\boldsymbol{\phi}'_k\boldsymbol{\phi}^T_k \tag{28}$$

$$\mathbb{E}\{e(\boldsymbol{\theta})\,\boldsymbol{\phi}\} \approx \mathbf{e}_{\boldsymbol{\Phi},t} = \frac{1}{t+1}\sum_{k=0}^{t}e_k\boldsymbol{\phi}_k \tag{29}$$

And obtain the iterative algorithm that we will identify with the acronym LPBR (linear prediction Bellman residual):

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t\left(\mathbf{e}_{\boldsymbol{\Phi},t} - \gamma\widehat{R}_{\boldsymbol{\Phi}'\boldsymbol{\Phi},t}\widehat{R}^{-1}_{\boldsymbol{\Phi},t}\mathbf{e}_{\boldsymbol{\Phi},t}\right) \tag{30}$$

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 [11] or TDC algorithm [9].

Our MDP is a Markov chain of 7 states [3], 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 1). 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.

For the simulations, 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 (26). The discount factor is set to $\gamma = 1$.

Some performance results are given in Figures 2.a and 2.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 (26), it still approximates the other statistics in (24) instantaneously. RLSTD is more accurate than TDC, and the proposed LPBR is even better. Note that, though the per-time complexity is $\mathcal{O}(n)$ in these algorithms, the less bias and variance of RLSTD and LPBR comes at the cost of more memory requirements, $\mathcal{O}(n^2)$, versus the linear memory requirements of of TDC.

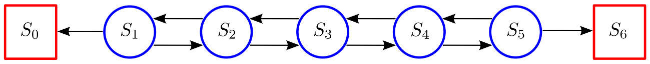
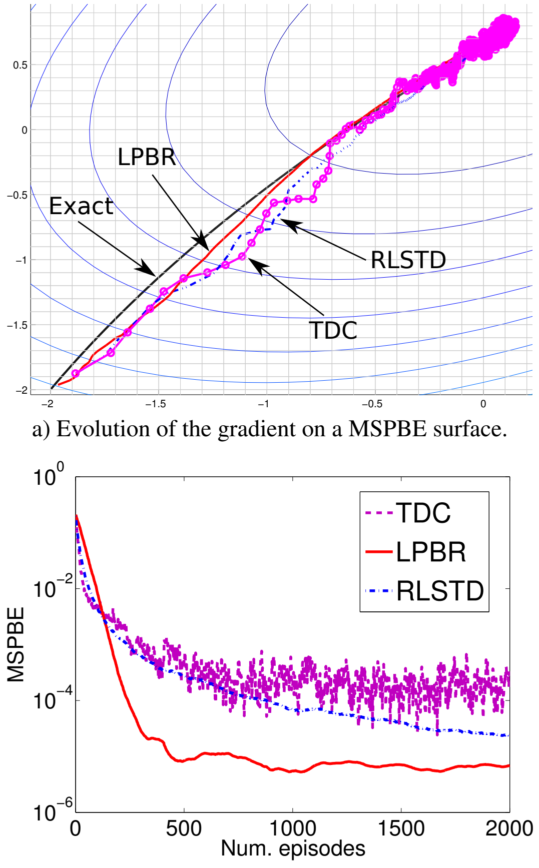


Fig. 1. State diagram of the random walk problem.



a) Evolution of the gradient on a MSPBE surface.

(b) Error curves

Fig. 2. Random walk in a Markov chain. (a) Evolution of the exact (18) and stochastic approximations (Algorithm 6 in [11], (25) and (30)) of the gradient, and (b) Error curves.

5. CONCLUSIONS AND FUTURE WORK

We have presented a fixed point solution for the 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, an efficient adaptive implementation that solves the double sampling requirement provides a fast and unbiased linear estimate.

This same approach could be extended in other directions, such as TD(λ) family of algorithms [12], off-policy iteration, distributed versions or multi-agent version.

It has to be noticed that we will just consider the policy evaluation problem in order to predict how good is a certain policy. As a future work, we will extend this algorithm for learning capabilities where the features will represent state-action pairs.

6. REFERENCES

[1] 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.

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

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

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

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

[10] 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.

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

[12] G.D. Konidaris, S. Niekum, and P.S. Thomas, "Td(gamma): Re-evaluating complex backups in temporal difference learning," in *Advances in Neural Information Processing Systems 24*, December 2011, pp. 2402–2410.