Recursive Least-Squares Off-policy Learning with Eligibility Traces

Bruno Scherrer, Matthieu Geist

Recursive Least-Squares Off-policy Learning with Eligibility Traces

Bruno Scherrer BRUNO.SCHERRER@INRIA.FR
MAIA, INRIA Lorraine, France

Matthieu Geist MATTHIEU.GEIST@SUPELEC.FR
UMI 2958 (Georgiatech-CNRS) Supélec, France

Abstract

In the framework of Markov Decision Processes, we consider the problem of learning a linear approximation of the value function of some fixed policy from one trajectory possibly generated by some other policy. We review *on-policy* learning least-squares algorithms of the literature (LSTD (Boyan, 1999), LSPE (Bertsekas and Ioffe, 1996), FPKF (Choi and Van Roy, 2006) and GPTD (Engel, 2005)/KTD (Geist and Pietquin, 2010b)). We then describe a systematic approach for adapting them to *off-policy* learning *with eligibility traces*. This leads to two known algorithms, LSTD(λ)/LSPE(λ) (Yu, 2010) and suggests new extensions of FPKF and GPTD/KTD. We describe their recursive implementation, discuss their convergence properties, and illustrate their behavior experimentally. Overall, our study suggests that the state-of-art LSTD(λ) (Yu, 2010) remains the best least-squares algorithm.

Keywords: Reinforcement Learning, Value Estimation, Linear Least-Squares Algorithms, Convergence Analysis

1. Introduction

We consider the problem of learning a linear approximation of the value function of some fixed policy in a Markov Decision Process (MDP) framework, in the most general situation where learning must be done from a single trajectory possibly generated by some other policy, also known as *off-policy* learning. Given samples, well-known methods for estimating a value function are temporal difference (TD) learning and Monte Carlo (Sutton and Barto, 1998). TD learning with eligibility traces (Sutton and Barto, 1998), known as TD(λ), constitutes a nice bridge between both approaches, and by controlling the bias/variance trade-off (Kearns and Singh, 2000), their use can significantly speed up learning. When the value function is approximated through a linear architecture, the depth λ of the eligibility traces is also known to control the quality of approximation (Tsitsiklis and Van Roy, 1997). Overall, the use of these traces often plays an important practical role.

In the *on-policy* case (where the policy to evaluate is the same as the one that generated data), there has been a significant amount of research on linear least-squares approaches, which are more sample-efficient than TD/Monte-Carlo. Such works include LSTD(λ) (Boyan, 1999), LSPE(λ) (Bertsekas and Ioffe, 1996), FPKF (Choi and Van Roy,

2006) and GPTD (Engel, 2005)/KTD (Geist and Pietquin, 2010b)[1]. Works on off-policy linear learning are sparser: Precup *et al.* (2000) proposed a variation of TD(λ) that can combine off-policy TD learning with linear approximation and eligibility traces. Recently, Yu (2010) proposed and analysed off-policy versions of LSTD(λ)/LSPE(λ). The first motivation of this article is to argue that it is conceptually simple to extend *all* the least-squares algorithms we have just mentionned so that they can be applied to the off-policy setting *and* use eligibility traces. If this allows to rederive the off-policy versions of LSTD(λ) and LSPE(λ) (Yu, 2010), it also leads to new candidate algorithms, for which we will derive recursive formulations. The second motivation of this work is to discuss the subtle differences between these intimately-related algorithms on the analytical side, and to provide some comparative insights on their empirical behavior (a topic that has to our knowledge not been considered in the literature, even in the simplest *on-policy* and *no-trace* situation).

The rest of the paper is organized as follows. Section 2 introduces the background of Markov Decision Processes and describes the state-of-the-art algorithms for on-policy learning with recursive least-squares methods. Section 3 shows how to adapt these methods so that they can both deal with the off-policy case and use eligibility traces. The resulting algorithms are formalized, the formula for their recursive implementation is derived, and we discuss their convergence properties. Section 4 illustrates empirically the behavior of these algorithms and Section 5 concludes.

2. Background and state-of-the-art on-policy algorithms

A Markov Decision Process (MDP) is a tuple $\{S, A, P, R, \gamma\}$ in which S is a finite state space identified with $\{1, 2, \ldots, N\}$, A a finite action space, $P \in \mathcal{P}(S)^{S\times A}$ the set of transition probabilities, $R \in \mathbb{R}^{S\times A}$ the reward function and γ the discount factor. A mapping $\pi \in \mathcal{P}(A)^S$ is called a policy. For any policy π, let P^π be the corresponding stochastic transition matrix, and R^π the vector of mean reward when following π, *i.e.*, of components $E_{a|\pi,s}[R(s,a)]$. The value $V^\pi(s)$ of state s for a policy π is the expected discounted cumulative reward starting in state s and then following the policy π:

$$V^\pi(s) = E_\pi\left[\sum_{i=0}^{\infty} \gamma^i r_i | s_0 = s\right],$$

where E_π denotes the expectation induced by policy π. The value function satisfies the (linear) Bellman equation:

$$\forall s, V^\pi(s) = E_{s',a|s,\pi}[R(s,a) + \gamma V^\pi(s')].$$

It can be rewritten as the fixed-point of the Bellman evaluation operator: $V^\pi = T^\pi V^\pi$ where for all V, $T^\pi V = R^\pi + \gamma P^\pi V$.

In this article, we are interested in learning an approximation of this value function V^π under some constraints. First, we assume our approximation to be linearly parameterized:

$$\hat{V}_\theta(s) = \theta^T \phi(s)$$

1. For the record, GPTD has been extended to the case $\lambda = 1$ (Engel *et al.*, 2005) and KTD to $\lambda \in [0, 1]$ (Geist and Pietquin, 2010a) but for a different (non-standard) notion of trace.

with $\theta \in \mathbb{R}^p$ being the parameter vector and $\phi(s)$ the feature vector. Also, we want to estimate the value function V^π (or equivalently the associated parameter θ) from a single finite trajectory generated using a possibly different behaviorial policy π_0. Let μ_0 be the stationary distribution of the stochastic matrix $P_0 = P^{\pi_0}$ of the *behavior policy* π_0 (we assume it exists and is unique). Let D_0 be the diagonal matrix of which the elements are $(\mu_0(s_i))_{1\leq i\leq N}$. Let Φ be the matrix of feature vectors:

$$\Phi = [\phi(1)\dots\phi(N)]^T.$$

The projection Π_0 onto the hypothesis space spanned by Φ with respect to the μ_0-quadratic norm, which will be central for the understanding of the algorithms, has the following closed-form:

$$\Pi_0 = \Phi(\Phi^T D_0 \Phi)^{-1}\Phi^T D_0.$$

In the rest of this section, we review existing on-policy least-squares based temporal difference learning algorithms. In this case, the behavior and target policies are the same so we omit the subscript 0 for the policy (π) and the projection (Π). We assume that a trajectory $(s_1, a_1,\ r_1, s_2, \dots,\ s_j, a_j, r_j,\ s_{j+1}, \dots s_{i+1})$ sampled according to the policy π is available. Let us introduce the sampled Bellman operator $\hat{T}_j$, defined as:

$$\hat{T}_j : V \in \mathbb{R}^S \to \hat{T}_j V = r_j + \gamma V(s_{j+1}) \in \mathbb{R}$$

so that $\hat{T}_j V$ is an unbiased estimate of $TV(s_j)$. If values were observable, estimating the projected parameter vector θ would reduce to project the value function onto the hypothesis space using the empirical projection operator. This would be the classical least-squares approach. Since values are not observed – only transitions (rewards and next states) are –, we will rely on *temporal differences* (terms of the form $\hat{T}_j V - V(s_j)$) to estimate the value function.

The Least-Squares Temporal Differences (LSTD) algorithm of Bradtke and Barto (1996) aims at finding the fixed-point of the operator being the composition of the projection onto the hypothesis space and of the Bellman operator. Otherwise speaking, it searches for the fixed-point $\hat{V}_\theta = \Pi T \hat{V}_\theta$, Π being the just introduced projection operator. Thus, using the available trajectory, LSTD solves the following fixed-point problem:

$$\theta_i = \operatorname*{argmin}_{\omega\in\mathbb{R}^p} \sum_{j=1}^{i} \left(\hat{T}_j \hat{V}_{\theta_i} - \hat{V}_\omega(s_j)\right)^2.$$

The Least-Squares Policy Evaluation (LSPE) algorithm of Bertsekas and Ioffe (1996) searches for the same fixed-point, but in an iterative way instead of directly (informally, $\hat{V}_{\theta_i} \simeq \Pi T \hat{V}_{\theta_{i-1}}$). The corresponding optimization problem is:

$$\theta_i = \operatorname*{argmin}_{\omega\in\mathbb{R}^p} \sum_{j=1}^{i} \left(\hat{T}_j \hat{V}_{\theta_{i-1}} - \hat{V}_\omega(s_j)\right)^2.$$

The Fixed-Point Kalman Filter (FPKF) algorithm of Choi and Van Roy (2006) is a least-squares variation of the classical temporal difference learning algorithm (Sutton and Barto,

1998). Value function approximation is treated as a supervised learning problem, and unobserved values are bootstrapped: the unobserved value $V^\pi(s_j)$ is replaced by the estimate $\hat{T}_j \hat{V}_{\theta_{j-1}}$. This is equivalent to solving the following optimization problem:

$$\theta_i = \operatorname*{argmin}_{\omega \in \mathbb{R}^p} \sum_{j=1}^{i} \left(\hat{T}_j \hat{V}_{\theta_{j-1}} - \hat{V}_\omega(s_j) \right)^2 .$$

Finally, the Bellman Residual Minimization algorithm aims at minimizing the distance between the value function and its image through the Bellman operator, $\|V - TV\|^2$. Notice that when the sampled operator is used, this leads to biased estimates (for instance, see Antos *et al.* (2006)). The corresponding optimization problem is as follows:

$$\theta_i = \operatorname*{argmin}_{\omega \in \mathbb{R}^p} \sum_{j=1}^{i} \left(\hat{T}_j \hat{V}_{\omega} - \hat{V}_\omega(s_j) \right)^2 .$$

This cost function has originally been proposed by Baird (1995) who minimized it using a stochastic gradient approach. It has been considered by Munos (2003) with a least-squares approach, however with a double sampling scheme to remove the bias. Both the parametric Gaussian Process Temporal Differences (GPTD) algorithm of Engel (2005) and the linear Kalman Temporal Differences (KTD) algorithm of Geist and Pietquin (2010b) can be shown to minimize the above cost using a least-squares approach (so with bias), and are thus the very same algorithm, that we will refer to as BRM (for "Bellman Residual Minimization") in the remaining of this paper.

To sum up, it appears that all the above mentionned algorithms of the literature minimize a cost-function of the form:

$$\theta_i = \operatorname*{argmin}_{\omega \in \mathbb{R}^p} \sum_{j=1}^{i} \left(\hat{T}_j \hat{V}_{\xi} - \hat{V}_\omega(s_j) \right)^2 . \tag{1}$$

Each of the presented approach is obtained by instantiating $\xi = \theta_i$, θ_{i-1}, θ_{j-1} or ω and solving the corresponding optimization problem. If more algorithms can be summarized under this generic equation (see Geist and Pietquin (2011)), the current paper focuses on linear least-squares based approaches.

3. Extension to eligibility traces and off-policy learning

This section contains the core of our contribution. We are going to describe a systematic approach in order to adapt the previously mentionned algorithms so that they can deal with eligibility traces and off-policy learning. The actual formalization of the algorithms, along with the derivation of their recursive implementation, will then follow.

Let $0 \leq \lambda \leq 1$ be the eligibility factor. Using eligibility traces amounts to looking for the fixed-point of the following variation of the Bellman operator (Bertsekas and Tsitsiklis, 1996):

$$\forall V \in \mathbb{R}^S, \quad T^\lambda V = (1-\lambda) \sum_{i=0}^{\infty} \lambda^i T^{i+1} V$$

that makes a geometric average with parameter λ of the powers of the original Bellman operator T. Clearly, any fixed-point of T is a fixed-point of T^λ and vice-versa. The equivalent writings (after some simple algebra, see for instanceNedić and Bertsekas (2003)):

$$\begin{aligned} T^\lambda V &= (I - \lambda\gamma P)^{-1}(R + (1-\lambda)\gamma PV) \\ &= V + (I - \lambda\gamma P)^{-1}(R + \gamma PV - V) \end{aligned} \tag{2}$$

lead to the following well-known *temporal difference* based expression

$$\forall s,\ T^\lambda V(s) = V(s) + E_\pi\left[\sum_{j=i}^{\infty}(\gamma\lambda)^{j-i}\Big(r_j + \gamma V(s_{j+1}) - V(s_j)\Big)\Big|s_i = s\right],$$

where we recall that E_π means that the expectation is done according to the target policy π. With $\lambda = 0$, we recover the Bellman evaluation equation. With $\lambda = 1$, this is the definition of the value function as the expected and discounted cumulative reward: $T^1V(s_i) = E_\pi[\sum_{j=i}^{\infty}\gamma^{j-i}r_j|s_i]$.

As learning is done over a finite trajectory, it is natural to introduce the following truncated operator, which considers samples until time n:

$$\forall s,\ T_n^\lambda V(s) = V(s) + E_\pi\left[\sum_{j=i}^{n}(\gamma\lambda)^{j-i}\Big(r_j + \gamma V(s_{j+1}) - V(s_j)\Big)\Big|s_i = s\right].$$

Assume from now on that we have a trajectory $(s_1, a_1, \ldots, s_j, a_j, r_j, s_{j+1}, \ldots s_{i+1})$ sampled according to some behaviour policy π_0. As the behaviour policy π_0 and the target policy π may be different, estimates of T_n^λ need to be corrected through importance sampling (Ripley, 1987). For all s, a, let us introduce the following weight:

$$\rho(s,a) = \frac{\pi(a|s)}{\pi_0(a|s)}.$$

In our trajectory context, write

$$\rho_i^j = \prod_{k=i}^{j}\rho_k \text{ with } \rho_j = \rho(s_j, a_j)$$

with the convention that if $j < i$, $\rho_i^j = 1$. Now, consider the off-policy, sampled and truncated $\hat{T}_{i,n}^\lambda : \mathbb{R}^S \to \mathbb{R}$ operator as:

$$\hat{T}_{i,n}^\lambda V = V(s_i) + \sum_{j=i}^{n}(\gamma\lambda)^{j-i}\Big(\rho_i^j\hat{T}_jV - \rho_i^{j-1}V(s_j)\Big).$$

It can be seen that $\hat{T}_{i,n}^\lambda V$ is an unbiased estimate of $T_n^\lambda V(s_i)$ (see Precup *et al.* (2000) and Yu (2010) for details).

By replacing $\hat{T}_j$ by $\hat{T}_{j,i}^\lambda$ in the optimization problem of Equation (1), we provide a generic way to extend most of the parametric value function approximators to eligibility traces in an off-policy manner:

$$\theta_i = \underset{\omega\in\mathbb{R}^p}{\operatorname{argmin}}\sum_{j=1}^{i}\left(\hat{T}_{j,i}^\lambda\hat{V}_\xi - \hat{V}_\omega(s_j)\right)^2.$$

In the rest of this section, by instantiating ξ to θ_i, θ_{i-1}, θ_{j-1} or ω, we derive the already existing algorithms off-policy LSTD(λ)/LSPE(λ) (Yu, 2010), and we extend two existing algorithms to eligibility traces and to off-policy learning, that we will naturally call FPKF(λ) and BRM(λ). With $\lambda = 0$, this exactly corresponds to the algorithms we described in the previous section. When $\lambda = 1$, it can be seen that

$$\hat{T}^{\lambda}_{i,n} V = \hat{T}^{1}_{i,n} V = \sum_{j=i}^{n} \gamma^{j-i} \rho_i^j r_j + \gamma^{n-i+1} \rho_i^n V(s_{n+1}).$$

Thus, if $\gamma^{n-i+1}\rho_i^n$ tends to 0 when n tends to infinity[2] so that the influence of ξ in the definition of $\hat{T}^{\lambda}_{i,n} V_\xi$ vanishes, all algorithms should asymptotically behave the same.

Recall that a linear parameterization is chosen here, $\hat{V}_\xi(s_i) = \xi^T \phi(s_i)$. We adopt the following notations:

$$\phi_i = \phi(s_i),\ \Delta\phi_i = \phi_i - \gamma\rho_i\phi_{i+1} \text{ and } \tilde{\rho}_j^{k-1} = (\gamma\lambda)^{k-j}\rho_j^{k-1}$$

The generic cost function to be solved is therefore:

$$\theta_i = \operatorname*{argmin}_{\omega \in \mathbb{R}^p} J(\omega;\xi) \quad \text{with} \quad J(\omega;\xi) = \sum_{j=1}^{i} (\phi_j^T \xi + \sum_{k=j}^{i} \tilde{\rho}_j^{k-1}(\rho_k r_k - \Delta\phi_k^T \xi) - \phi_j^T \omega)^2. \qquad (3)$$

Before deriving existing and new algorithms, as announced, some required useful lemma are provided.

3.1 Some useful lemma

The first lemma allows computing directly the inverse of a rank-one perturbated matrix.

Lemma 1 (Sherman-Morrison) *Assume that A is an invertible $n \times n$ matrix and that $u, v \in \mathbb{R}^n$ are two vectors satisfying $1 + v^T A^{-1} u \neq 0$. Then:*

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^T A^{-1}}{1 + v^T A^{-1} u}$$

The second one is the Woodbury matrix identity which generalizes the Sherman-Morrison formula:

Lemma 2 (Woodbury) *Let A, U, C and V be matrices of correct sizes, then:*

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$$

The next lemma is simply a rewriting of imbricated sums. However, it is quite important here as it will allow stepping from the anti-causal operator $\hat{T}^{\lambda}_{i,j}$ – forward view of eligibility traces – to the causal recursion over parameters – backward view of eligibility traces – (see Sutton and Barto (1998) for discussions on backward/forward views).

2. This is not always the case, see Yu (2010) and the discussion in Section 3.5.

Lemma 3 *Let $f \in \mathbb{R}^{\mathbb{N}\times\mathbb{N}}$ and $n \in \mathbb{N}$. We have:*

$$\sum_{i=1}^{n}\sum_{j=i}^{n} f(i,j) = \sum_{i=1}^{n}\sum_{j=1}^{i} f(j,i)$$

The last lemma is also a rewriting of imbricated sums:

Lemma 4 *Let $f \in \mathbb{R}^{\mathbb{N}\times\mathbb{N}\times\mathbb{N}}$ and $n \in \mathbb{N}$. We have:*

$$\sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=i}^{n} f(i,j,k) = \sum_{i=1}^{n}\sum_{j=1}^{i}\sum_{k=1}^{j} f(k,i,j) + \sum_{i=2}^{n}\sum_{j=1}^{i-1}\sum_{k=1}^{j} f(k,j,i)$$

3.2 Off-policy LSTD(λ)

The off-policy LSTD(λ) algorithm corresponds to instantiating Problem (3) with $\xi = \theta_i$:

$$\theta_i = \underset{\omega\in\mathbb{R}^p}{\operatorname{argmin}} \sum_{j=1}^{i} (\phi_j^T\theta_i + \sum_{k=j}^{i} \tilde{\rho}_j^{k-1}(\rho_k r_k - \Delta\phi_k^T\theta_i) - \phi_j^T\omega)^2.$$

This can be solved by zeroing the gradient respectively to ω:

$$\begin{aligned} \theta_i &= (\sum_{j=1}^{i}\phi_j\phi_j^T)^{-1}\sum_{j=1}^{i}\phi_j(\phi_j^T\theta_i + \sum_{k=j}^{i}\tilde{\rho}_j^{k-1}(\rho_k r_k - \Delta\phi_k^T\theta_i)) \\ \Leftrightarrow \quad 0 &= \sum_{j=1}^{i}\sum_{k=j}^{i}\phi_j\tilde{\rho}_j^{k-1}(\rho_k r_k - \Delta\phi_k^T\theta_i), \end{aligned}$$

which, through Lemma 3, is equivalent to:

$$0 = \sum_{j=1}^{i}(\sum_{k=1}^{j}\phi_k\tilde{\rho}_k^{j-1})(\rho_j r_j - \Delta\phi_j^T\theta_i).$$

Introducing the (corrected) eligibility vector z_j:

$$z_j = \sum_{k=1}^{j}\phi_k\tilde{\rho}_k^{j-1} = \sum_{k=1}^{j}\phi_k(\gamma\lambda)^{j-k}\prod_{m=k}^{j-1}\rho_m = \gamma\lambda\rho_{j-1}z_{j-1} + \phi_j, \tag{4}$$

one obtains the following batch estimate:

$$\theta_i = (\sum_{j=1}^{i} z_j\Delta\phi_j^T)^{-1}\sum_{j=1}^{i} z_j\rho_j r_j = (A_i)^{-1}b_i \tag{5}$$

where

$$A_i = \sum_{j=1}^{i} z_j\Delta\phi_j^T \quad \text{and} \quad b_i = \sum_{j=1}^{i} z_j\rho_j r_j. \tag{6}$$

Thanks to Lemma 1, the inverse $M_i = (A_i)^{-1}$ can be computed recursively:

$$M_i = (\sum_{j=1}^{i} z_j \Delta\phi_j^T)^{-1} = M_{i-1} - \frac{M_{i-1} z_i \Delta\phi_i^T M_{i-1}}{1 + \Delta\phi_i^T M_{i-1} z_i}.$$

This can be used to derive a recursive estimate:

$$\begin{aligned}\theta_i &= (\sum_{j=1}^{i} z_j \Delta\phi_j^T)^{-1} \sum_{j=1}^{i} z_j \rho_j r_j = (M_{i-1} - \frac{M_{i-1} z_i \Delta\phi_i^T M_{i-1}}{1 + \Delta\phi_i^T M_{i-1} z_i})(\sum_{j=1}^{i-1} z_j r_j \rho_j + z_i \rho_i r_i) \\ &= \theta_{i-1} + \frac{M_{i-1} z_i}{1 + \Delta\phi_i^T M_{i-1} z_i}(\rho_i r_i - \Delta\phi_i^T \theta_{i-1}).\end{aligned}$$

Writing K_i the gain $\frac{M_{i-1} z_i}{1+\Delta\phi_i^T M_{i-1} z_i}$, this gives Alg. 1.

Algorithm 1: Off-policy LSTD(λ)

Initialization;
Initialize vector θ_0 and matrix M_0 ;
Set $z_0 = 0$;

for $i = 1, 2, \ldots$ **do**

Observe ϕ_i, r_i, ϕ_{i+1} ;

Update traces ;
$z_i = \gamma\lambda\rho_{i-1} z_{i-1} + \phi_i$;

Update parameters ;
$K_i = \frac{M_{i-1} z_i}{1+\Delta\phi_i^T M_{i-1} z_i}$;
$\theta_i = \theta_{i-1} + K_i(\rho_i r_i - \Delta\phi_i^T \theta_{i-1})$;
$M_i = M_{i-1} - K_i (M_{i-1}^T \Delta\phi_i)^T$;

This algorithm has been proposed and analyzed recently by Yu (2010). The author proves the following result: if the *behavior* policy π_0 induces an irreducible Markov chain and chooses with positive probability any action that may be chosen by the *target* policy π, and if the compound (linear) operator $\Pi_{\pi_0} T^\lambda$ has a unique fixed-point[3], then off-policy LSTD(λ) converges to it almost surely. Formally, it converges to the solution θ^* of the so-called *projected fixed-point* equation:

$$V_{\theta^*} = \Pi_0 T^\lambda V_{\theta^*}. \tag{7}$$

Using the expression of the projection Π_0 and the form of the Bellman operator in Equation (2), it can be seen that θ^* satisfies (see Yu (2010) for details)

$$\theta^* = A^{-1} b$$

where

$$A = \Phi^T D_0 (I - \gamma P)(I - \lambda\gamma P)^{-1} \Phi \quad \text{and} \quad b = \Phi^T D_0 (I - \lambda\gamma P)^{-1} R. \tag{8}$$

3. It is not always the case, see Tsitsiklis and Van Roy (1997) or Section 4 for a counter-example.

The core of the analysis of Yu (2010) consists in showing that $\frac{1}{i}A_i$ and $\frac{1}{i}b_i$ defined in Equation (6) respectively converge to A and b almost surely. Through Equation (5), this implies the convergence of θ_i to θ^*.

3.3 Off-policy LSPE(λ)

The off-policy LSPE(λ) algorithm corresponds to the instantiation $\xi = \theta_{i-1}$ in Problem (3):

$$\theta_i = \operatorname*{argmin}_{\omega \in \mathbb{R}^p} \sum_{j=1}^{i} (\phi_j^T \theta_{i-1} + \sum_{k=j}^{i} \tilde{\rho}_j^{k-1}(\rho_k r_k - \Delta\phi_k^T \theta_{i-1}) - \phi_j^T \omega)^2.$$

This can be solved by zeroing the gradient respectively to ω:

$$\begin{aligned}\theta_i &= (\sum_{j=1}^{i} \phi_j \phi_j^T)^{-1} \sum_{j=1}^{i} \phi_j (\phi_j^T \theta_{i-1} + \sum_{k=j}^{i} \tilde{\rho}_j^{k-1}(\rho_k r_k - \Delta\phi_k^T \theta_{i-1})) \\ &= \theta_{i-1} + (\sum_{j=1}^{i} \phi_j \phi_j^T)^{-1} \sum_{j=1}^{i} \sum_{k=j}^{i} \phi_j \tilde{\rho}_j^{k-1}(\rho_k r_k - \Delta\phi_k^T \theta_{i-1}).\end{aligned}$$

Lemma 3 can be used (recall the definition of the eligibility vector z_j in Equation (4)):

$$\begin{aligned}\theta_i &= \theta_{i-1} + (\sum_{j=1}^{i} \phi_j \phi_j^T)^{-1} \sum_{j=1}^{i} \sum_{k=1}^{j} \phi_k \tilde{\rho}_k^{j-1}(\rho_j r_j - \Delta\phi_j^T \theta_{i-1}) \\ &= \theta_{i-1} + (\sum_{j=1}^{i} \phi_j \phi_j^T)^{-1} \sum_{j=1}^{i} z_j (\rho_j r_j - \Delta\phi_j^T \theta_{i-1}).\end{aligned}$$

Define the matrix N_i as follows:

$$N_i = (\sum_{j=1}^{i} \phi_j \phi_j^T)^{-1} = N_{i-1} - \frac{N_{i-1}\phi_i \phi_i^T N_{i-1}}{1 + \phi_i^T N_{i-1} \phi_i}, \tag{9}$$

where the second equality follows from Lemma 1. Let A_i and b_i be defined as in the LSTD description in Equation (6). For clarity, we restate their definition along with their recursive writing:

$$\begin{aligned}A_i &= \sum_{j=1}^{i} z_j \Delta\phi_j^T = A_{i-1} + z_i \Delta\phi_{i+1}^T \\ b_i &= \sum_{j=1}^{i} z_j \rho_j r_j = b_{i-1} + z_i \rho_i r_i.\end{aligned}$$

Then, it can be seen that the LSPE(λ) update is:

$$\theta_i = \theta_{i-1} + N_i(b_i - A_i \theta_{i-1}).$$

The overall computation is provided in Alg. 2. This algorithm, (briefly) mentionned by

Algorithm 2: Off-policy LSPE(λ)

Initialization;
Initialize vector θ_0 and matrix N_0 ;
Set $z_0 = 0$, $A_0 = 0$ and $b_0 = 0$;

for $i = 1, 2, \ldots$ **do**

Observe ϕ_i, r_i, ϕ_{i+1};

Update traces ;
$z_i = \gamma\lambda\rho_{i-1} z_{i-1} + \phi_i$;

Update parameters ;
$N_i = N_{i-1} - \frac{N_{i-1}\phi_i \phi_i^T N_{i-1}}{1+\phi_i^T N_{i-1}\phi_i}$;
$A_i = A_{i-1} + z_i \Delta\phi_i^T$;
$b_i = b_{i-1} + \rho_i z_i r_i$;
$\theta_i = \theta_{i-1} + N_i(b_i - A_i\theta_{i-1})$;

Yu (2010), generalizes the LSPE(λ) algorithm of Bertsekas and Ioffe (1996) to off-policy learning. With respect to LSTD(λ), which computes $\theta_i = (A_i)^{-1} b_i$ (*cf.* Equation (5)) at each iteration, LSPE(λ) is fundamentally recursive. Along with the almost sure convergence of $\frac{1}{i}A_i$ and $\frac{1}{i}b_i$ to A and b (defined in Equation (8)), it can be shown that iN_i converges to $N = (\Phi^T D_0 \Phi)^{-1}$ (see for instance Nedić and Bertsekas (2003)) so that, asymptotically, LSPE(λ) behaves as:

$$\theta_i = \theta_{i-1} + N(b - A\theta_{i-1}) = Nb + (I - NA)\theta_{i-1}$$

or using the defintion of Π_0, A, b (Equation (8)) and T^λ (Equation (2)):

$$V_{\theta_i} = \Phi\theta_i = \Phi Nb + \Phi(I - NA)\theta_{i-1} = \Pi_0 T^\lambda V_{\theta_{i-1}}. \tag{10}$$

The behavior of this sequence depends on whether the spectral radius of $\Pi_0 T^\lambda$ is smaller than 1 or not. Thus, the analyses of Yu (2010) and Nedić and Bertsekas (2003) (for the convergence of N_i) imply the following convergence result[4]: under the assumptions required for the convergence of off-policy LSTD(λ), and the additional assumption that the operator $\Pi_0 T^\lambda$ has a spectral radius smaller than 1 (so that it is contracting), LSPE(λ) also converges almost surely to the fixed-point of the compound $\Pi_0 T^\lambda$ operator.

There are two sufficient conditions that can ensure such a desired contraction property. The first one is when one considers on-policy learning, as Nedić and Bertsekas (2003) did when they derived the first convergence proof of (on-policy) LSPE(λ). When the behavior policy π_0 is different from the target policy π, a sufficient condition for contraction is that λ be close enough to 1; indeed, when λ tends to 1, the spectral radius of T^λ tends to zero and can potentially balance an expansion of the projection Π_0. In the off-policy case, when γ is sufficiently big, a small value of λ can make $\Pi_0 T^\lambda$ expansive (see Tsitsiklis and Van Roy

4. Though it is not stated explicitely there, the credit of this convergence result should be given to Yu (2010), whose analysis allows to easily conclude.

(1997) for an example in the case $\lambda = 0$) and off-policy LSPE(λ) will then diverge. Eventually, Equations (7) and (10) show that when $\lambda = 1$, both LSTD(λ) and LSPE(λ) asymptotically coincide (as $T^1 V$ does not depend on V).

3.4 Off-policy FPKF(λ)

The off-policy FPKF(λ) algorithm corresponds to the instantiation $\xi = \theta_{j-1}$ in Problem (3):

$$\theta_i = \underset{\omega \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{j=1}^{i} (\phi_j^T \theta_{j-1} + \sum_{k=j}^{i} \tilde{\rho}_j^{k-1} (\rho_k r_k - \Delta\phi_k^T \theta_{j-1}) - \phi_j^T \omega)^2.$$

This can be solved by zeroing the gradient respectively to ω:

$$\theta_i = N_i \sum_{j=1}^{i} \phi_j (\phi_j^T \theta_{j-1} + \sum_{k=j}^{i} \tilde{\rho}_j^{k-1} (\rho_k r_k - \Delta\phi_k^T \theta_{j-1})),$$

where N_i is the matrix introduced for LSPE(λ) in Equation (9). For clarity, we restate its definition here and its recursive writing:

$$N_i = (\sum_{j=1}^{i} \phi_j \phi_j^T)^{-1} = N_{i-1} - \frac{N_{i-1} \phi_i \phi_i^T N_{i-1}}{1 + \phi_i^T N_{i-1} \phi_i}. \tag{11}$$

Using Lemma 3, one obtains:

$$\theta_i = N_i (\sum_{j=1}^{i} \phi_j \phi_j^T \theta_{j-1} + \sum_{j=1}^{i} \sum_{k=1}^{j} \phi_k \tilde{\rho}_k^{j-1} (\rho_j r_j - \Delta\phi_j^T \theta_{k-1})).$$

With respect to the previously described algorithms, the difficulty here is that on the right side there is a dependence with all the previous terms θ_{k-1} for $1 \leq k \leq i$. Using the symmetry of the dot product $\Delta\phi_j^T \theta_{k-1} = \theta_{k-1}^T \Delta\phi_j$, it is possible to write a recursive algorithm by introducing the trace matrix Z_j that integrates the subsequent values of θ_k as follows:

$$Z_j = \sum_{k=1}^{j} \tilde{\rho}_k^{j-1} \phi_k \theta_{k-1}^T = Z_{j-1} + \gamma\lambda\rho_{j-1} \phi_j \theta_{j-1}^T.$$

With this notation we obtain:

$$\theta_i = N_i (\sum_{j=1}^{i} \phi_j \phi_j^T \theta_{j-1} + \sum_{j=1}^{i} (z_j \rho_j r_j - Z_j \Delta\phi_j).$$

Using Equation (11) and a few algebraic manipulations, we end up with:

$$\theta_i = \theta_{i-1} + N_i (z_i \rho_i r_i - Z_i \Delta\phi_i).$$

This is the parameter update as provided in Alg. 3. It generalizes the FPKF algorithm of Choi and Van Roy (2006) that was originally only introduced without traces and in the on-policy case. As LSPE(λ), this algorithm is fundamentally recursive. However, its

Algorithm 3: Off-policy FPKF(λ)

Initialization;
Initialize vector θ_0 and matrix N_0 ;
Set $z_0 = 0$ and $Z_0 = 0$;

for $i = 1, 2, \ldots$ **do**

Observe ϕ_i, r_i, ϕ_{i+1};

Update traces ;
$z_i = \gamma\lambda\rho_{i-1} z_{i-1} + \phi_i$;
$Z_i = \gamma\lambda\rho_{i-1} Z_{i-1} + \phi_i \theta_{i-1}^T$;

Update parameters ;
$N_i = N_{i-1} - \frac{N_{i-1}\phi_i \phi_i^T N_{i-1}}{1+\phi_i^T N_{i-1}\phi_i}$;
$\theta_i = \theta_{i-1} + N_i(z_i\rho_i r_i - Z_i\Delta\phi_i)$;

overall behavior is quite different. As we discussed for LSPE(λ), iN_i can be shown to tend asymptotically $N = (\Phi^T D_0 \Phi)^{-1}$ and FPKF(λ) iterates eventually resemble:

$$\theta_i = \theta_{i-1} + \frac{1}{i} N(z_i\rho_i r_i - Z_i\Delta\phi_i).$$

The term in brackets is a random component (that depends on the last transition) and $\frac{1}{i}$ acts as a learning coefficient that asymptotically tends to 0. In other words, FPKF(λ) has a *stochastic approximation* flavour. In particular, one can see FPKF(0) as a stochastic approximation of LSPE(0). Indeed, asymptotically, FPKF(0) does

$$\theta_i = \theta_{i-1} + \frac{1}{i} N(\rho_i\phi_i r_i - \phi_i\Delta\phi_i^T\theta_{i-1}),$$

and one can notice that $\rho_i\phi_i r_i$ and $\phi_i\Delta\phi_i^T$ are samples of A and b to which A_i and b_i converge through LSPE(0). When $\lambda > 0$, the situation is less clear (all the more that, as previously mentionned, we expect LSTD/LSPE/FPKF to asymptotically behave the same when λ tends to 1).

Due to its much more involved form (notably the matrix trace Z_j integrating the values of all the values θ_k from the start), we have not been able to obtain a formal analysis of FPKF(λ), even in the on-policy case. To our knowledge, there does not exist any *proof of convergence* for stochastic approximation algorithms in the off-policy case[5], and a related result for FPKF(λ) thus seems difficult. Based on the above-mentionned relation between FPKF(0) and LSPE(0) and the experiments we have run (see Section 4), we conjecture that off-policy FPKF(λ) has the same asymptotic behavior as LSPE(λ).

5. An analysis of TD(λ), with a simplifying assumption that forces the algorithm to stay bounded is given in Yu (2010). An analysis of a related algorithm, GQ(λ), is provided in Maei and Sutton (2010), with an assumption on the second moment of the traces, which does not hold in general (see Propostion 2 in Yu (2010)). A full analysis of these algorithms thus remains to be done.

3.5 Off-policy BRM(λ)

The off-policy BRM(λ) algorithm corresponds to the instantiation $\xi = \omega$ in Problem (3):

$$\theta_i = \operatorname*{argmin}_{\omega\in\mathbb{R}^p} \sum_{j=1}^{i} (\phi_j^T\omega + \sum_{k=j}^{i} \tilde{\rho}_j^{k-1}(\rho_k r_k - \Delta\phi_k^T\omega) - \phi_j^T\omega)^2 = \operatorname*{argmin}_{\omega\in\mathbb{R}^p} \sum_{j=1}^{i} (\sum_{k=j}^{i} \tilde{\rho}_j^{k-1}(\rho_k r_k - \Delta\phi_k^T\omega))^2.$$

Define

$$\psi_{j\to i} = \sum_{k=j}^{i} \tilde{\rho}_j^{k-1}\Delta\phi_k$$

$$\text{and} \quad z_{j\to i} = \sum_{k=j}^{i} \tilde{\rho}_j^{k-1}\rho_k r_k.$$

This yields to the following batch estimate:

$$\theta_i = \operatorname*{argmin}_{\omega\in\mathbb{R}^p} \sum_{j=1}^{i} (z_{j\to i} - \psi_{j\to i}^T\omega)^2 = (\tilde{A}_i)^{-1}\tilde{b}_i$$

where

$$\tilde{A}_i = \sum_{j=1}^{i} \psi_{j\to i}\psi_{j\to i}^T \quad \text{and} \quad \tilde{b}_i = \sum_{j=1}^{i} \psi_{j\to i} z_{j\to i}.$$

To obtain a recursive formula, these two sums have to be reworked through Lemma 4. Let us first focus on the latter:

$$\begin{aligned}\sum_{j=1}^{i} \psi_{j\to i} z_{j\to i} &= \sum_{j=1}^{i}\sum_{k=j}^{i}\sum_{m=j}^{i} \tilde{\rho}_j^{k-1}\Delta\phi_k\tilde{\rho}_j^{m-1}\rho_m r_m \\ &= \sum_{j=1}^{i}\sum_{k=1}^{j}\sum_{m=1}^{k} \tilde{\rho}_m^{j-1}\Delta\phi_j\tilde{\rho}_m^{k-1}\rho_k r_k + \sum_{j=2}^{i}\sum_{k=1}^{j-1}\sum_{m=1}^{k} \tilde{\rho}_m^{k-1}\Delta\phi_k\tilde{\rho}_m^{j-1}\rho_j r_j.\end{aligned}$$

Writing

$$y_k = \sum_{m=1}^{k} (\tilde{\rho}_m^{k-1})^2 = 1 + (\gamma\lambda\rho_{k-1})^2 y_{k-1},$$

we have that:

$$\sum_{m=1}^{k} \tilde{\rho}_m^{j-1}\tilde{\rho}_m^{k-1} = \tilde{\rho}_k^{j-1} y_k.$$

Therefore:

$$\sum_{j=1}^{i} \psi_{j\to i} z_{j\to i} = \sum_{j=1}^{i}\sum_{k=1}^{j} \tilde{\rho}_k^{j-1} y_k \Delta\phi_j\rho_k r_k + \sum_{j=2}^{i}\sum_{k=1}^{j-1} \tilde{\rho}_k^{j-1} y_k \Delta\phi_k\rho_j r_j.$$

With the following notations:

$$z_j = \sum_{k=1}^{j} \tilde{\rho}_k^{j-1} y_k \rho_k r_k = \gamma\lambda\rho_{j-1} z_{j-1} + \rho_j r_j y_j$$
$$\text{and} \quad \Delta_j = \sum_{k=1}^{j} \tilde{\rho}_k^{j-1} y_k \Delta\phi_k = \gamma\lambda\rho_{j-1}\Delta_{j-1} + y_j \Delta\phi_j,$$

and with the convention that $z_0 = 0$ and $\Delta_0 = 0$, one can write:

$$\sum_{j=1}^{i} \psi_{j\to i} z_{j\to i} = \sum_{j=1}^{i} (\Delta\phi_j \rho_j r_j y_j + \gamma\lambda\rho_{j-1}(\Delta\phi_j z_{j-1} + \rho_j r_j \Delta_{j-1}))$$

Similarly, on can show that:

$$\sum_{j=1}^{i} \psi_{j\to i}\psi_{j\to i}^T = \sum_{j=1}^{i} (\Delta\phi_j \Delta\phi_j^T y_j + \gamma\lambda\rho_{j-1}(\Delta\phi_j \Delta_{j-1}^T + \Delta_{j-1}\Delta\phi_j^T))$$

Denoting

$$u_j = \sqrt{y_j}\Delta\phi_j,$$
$$v_j = \frac{\gamma\lambda\rho_{j-1}}{\sqrt{y_j}}\Delta_{j-1},$$

and I_2 the 2×2 identity matrix, we have:

$$\begin{aligned}\sum_{j=1}^{i} \psi_{j\to i}\psi_{j\to i}^T &= \sum_{j=1}^{i} ((u_j + v_j)(u_j + v_j)^T - v_j v_j^T) \\ &= \sum_{j=1}^{i-1} \psi_{j\to i}\psi_{j\to i}^T + \underbrace{\begin{pmatrix} u_i + v_i & v_i \end{pmatrix}}_{=U_i} I_2 \underbrace{\begin{pmatrix} (u_i + v_i)^T \\ -v_i^T \end{pmatrix}}_{=V_i}.\end{aligned}$$

We can apply the Woodbury identity given in Lemma 2:

$$\begin{aligned}C_i = \left(\sum_{j=1}^{i} \psi_{j\to i}\psi_{j\to i}^T\right)^{-1} &= \left(\sum_{j=1}^{i-1} \psi_{j\to i} z_{j\to i} + U_i I_2 V_i\right)^{-1} \\ &= C_{i-1} - C_{i-1}U_i \left(I_2 + V_i C_{i-1} U_i\right)^{-1} V_i C_{i-1}.\end{aligned}$$

The other sum can also be reworked:

$$\begin{aligned}\tilde{b}_i = \sum_{j=1}^{i} \psi_{j\to i} z_{j\to i} &= \sum_{j=1}^{i} \Delta\phi_j r_j y_j + \gamma\lambda\left(\Delta_{j-1} r_j + \Delta\phi_j z_{j-1}\right) \\ &= \tilde{b}_{i-1} + \Delta\phi_i r_i y_i + \gamma\lambda\left(\Delta_{i-1} r_i + \Delta\phi_i z_{i-1}\right) = \tilde{b}_{i-1} + U_i \underbrace{\begin{pmatrix} \sqrt{y_i} r_i + \frac{\gamma\lambda}{\sqrt{y_i}} z_{i-1} \\ -\frac{\gamma\lambda}{\sqrt{y_i}} z_{i-1} \end{pmatrix}}_{=W_i}.\end{aligned}$$

Finally, the recursive BRM(λ) estimate can be computed as follows:

$$\theta_i = C_i \tilde{b}_i = \theta_{i-1} + C_{i-1} U_i \left(I_2 + V_i C_{i-1} U_i\right)^{-1} \left(W_i - V_i \theta_{i-1}\right).$$

The matrix to be inverted being a 2×2 matrix, it admits a straightforward analytical solution. This gives BRM(λ) as provided in Alg. 4.

Algorithm 4: Off-policy BRM(λ)

Initialization;
Initialize vector θ_0 and matrix C_0 ;
Set $y_0 = 0$, $\Delta_0 = 0$ and $z_0 = 0$;

for $i = 1, 2, \ldots$ **do**

Observe ϕ_i, r_i, ϕ_{i+1};

Pre-update traces ;
$y_i = (\gamma\lambda\rho_{i-1})^2 y_{i-1} + 1$;

Compute ;
$U_i = \begin{pmatrix} \sqrt{y_i}\Delta\phi_i + \frac{\gamma\lambda\rho_{i-1}}{\sqrt{y_i}}\Delta_{i-1} & \frac{\gamma\lambda\rho_{i-1}}{\sqrt{y_i}}\Delta_{i-1} \end{pmatrix}$;
$V_i = \begin{pmatrix} \sqrt{y_i}\Delta\phi_i + \frac{\gamma\lambda\rho_{i-1}}{\sqrt{y_i}}\Delta_{i-1} & -\frac{\gamma\lambda\rho_{i-1}}{\sqrt{y_i}}\Delta_{i-1} \end{pmatrix}^T$;
$W_i = \begin{pmatrix} \sqrt{y_i}\rho r_i + \frac{\gamma\lambda\rho_{i-1}}{\sqrt{y_i}} z_{i-1} & -\frac{\gamma\lambda\rho_{i-1}}{\sqrt{y_i}} z_{i-1} \end{pmatrix}^T$;

Update parameters ;
$\theta_i = \theta_{i-1} + C_{i-1} U_i \left(I_2 + V_i C_{i-1} U_i\right)^{-1} \left(W_i - V_i \theta_{i-1}\right)$;
$C_i = C_{i-1} - C_{i-1} U_i \left(I_2 + V_i C_{i-1} U_i\right)^{-1} V_i C_{i-1}$;

Post-update traces ;
$\Delta_i = (\gamma\lambda\rho_{i-1})\Delta_{i-1} + \Delta\phi_i y_i$;
$z_i = (\gamma\lambda\rho_{i-1}) z_{i-1} + r_i \rho_i y_i$;

GPTD and KTD, which are close to BRM, have also been extended with some trace mechanism; however, GPTD(λ) (Engel, 2005), KTD(λ) (Geist and Pietquin, 2010a) and the just described BRM(λ) are different algorithms. Briefly, GPTD(λ) is very close to LSTD(λ) (in particular, GPTD(0) differs from GPTD) and KTD(λ) uses a different Bellman operator[6]. As BRM(λ) builds a linear systems of which it updates the solution recursively, it resembles LSTD(λ). However, the system it builds is different. The following theorem characterizes the behavior of BRM(λ) and its potential limit.

Theorem 5 *Assume that the stochastic matrix P_0 of the* behavior *policy is irreducible and has stationary distribution μ_0. Further assume that there exists a coefficient $\beta < 1$ such that*

$$\forall (s,a), \quad \lambda\gamma\rho(s,a) \leq \beta, \tag{12}$$

6. Actually, the corresponding loss is $(\hat{T}^0_{j,i}\hat{V}(\omega) - \hat{V}_\omega(s_j) + \gamma\lambda(\hat{T}^1_{j+1,i}\hat{V}(\omega) - \hat{V}_\omega(s_{j+1})))^2$. With $\lambda = 0$ it gives $\hat{T}^0_{j,i}$ and with $\lambda = 1$ it provides $\hat{T}^1_{j,i}$

then $\frac{1}{i}\tilde{A}_i$ *and* $\frac{1}{i}\tilde{b}_i$ *respectively converge almost surely to*

$$\tilde{A} = \Phi^T \left[D - \gamma DP - \gamma P^T D + \gamma^2 D' + S(I - \gamma P) + (I - \gamma P^T) S^T \right] \Phi$$
$$\tilde{b} = \Phi^T \left[(I - \gamma P^T) Q^T D + S \right] R^\pi$$

where we wrote:

$$D = \text{diag}\left((I - (\lambda\gamma)^2 \tilde{P}^T)^{-1} \mu_0 \right) \qquad Q = (I - \lambda\gamma P)^{-1}$$
$$D' = \text{diag}\left(\tilde{P}^T (I - (\lambda\gamma)^2 \tilde{P}^T)^{-1} \mu_0 \right) \qquad S = \lambda\gamma(DP - \gamma D')Q$$

and where $\tilde{P}$ *is the matrix of which the coordinates are* $\tilde{p}_{ss'} = \sum_a \pi(s,a)\rho(s,a)T(s,a,s')$. *As a consequence the BRM(*λ*) algorithm converges with probability 1 to* $\tilde{A}^{-1}\tilde{b}$.

The assumption given by Equation (12) trivially holds in the on-policy case (in which $\rho(s,a) = 1$ for all (s,a)) and in the off-policy case when $\lambda\gamma$ is sufficiently small with respect to the mismatch between policies. Note in particular that our result implies the almost sure convergence of the GPTD/KTD algorithms in the on-policy and no-trace case, a question that was still open in the literature (see for instance the conclusion of Engel (2005)). The matrix $\tilde{P}$, which is in general not a stochastic matrix, can have a spectral radius bigger than 1; Equation (12) ensures that $(\lambda\gamma)^2\tilde{P}$ has a spectral radius smaller than β so that D and D' are well defined. Removing assumption of Equation (12) does not seem easy, since by tuning $\lambda\gamma$ maliciously, one may force the spectral radius of $(\lambda\gamma)^2\tilde{P}$ to be as close to 1 as one may want, which would make $\tilde{A}$ and $\tilde{b}$ diverge. Though the quantity $\tilde{A}^{-1}\tilde{b}$ may compensate for these divervgence, our current proof technique cannot account for this situation and a related analysis constitutes possible future work.

In a bit more details, the proof of this Theorem is similar to that of Proposition 4 in Bertsekas and Yu (2009) and is detailed in the Appendix. The overall arguments are the following: Equation (12) implies that the traces can be truncated at some depth l, of which the influence on the potential limit of the algorithm vanishes when l tends to ∞. For all l, the l-truncated version of the algorithm can easily be analyzed through the ergodic theorem for Markov chains. Making l tend to ∞ allows to tie the convergence of the original arguments to that of the truncated version. Eventually, the formula for the limit of the truncated algorithm is computed and one derives the limit.

The fundamental idea behind the Bellman Residual approach is to address the computation of the fixed-point of T^λ differently from the previous methods. Instead of computing the projected fixed-point as in Equation (7), one considers the overdetermined system (and use Equation (2))

$$\begin{aligned} & \Phi\theta \simeq T^\lambda \Phi\theta \\ \Leftrightarrow \quad & \Phi\theta \simeq (I - \lambda\gamma P)^{-1}(R + (1-\lambda)\gamma P \Phi\theta) \\ \Leftrightarrow \quad & \Phi\theta \simeq QR + (1-\lambda)\gamma PQ\Phi\theta \\ \Leftrightarrow \quad & \Psi\theta \simeq QR \end{aligned}$$

with $\Psi = \Phi - (1-\lambda)\gamma PQ\Phi$, and solves it in a least-squares sense, that is by computing $\theta^* = \bar{A}^{-1}\bar{b}$ with $\bar{A} = \Psi^T\Psi$ and $\bar{b} = \Psi^T QR$. One of the motivation for this approach is

that, contrary to the matrix A of LSTD/LSPE/FPKF, $\bar{A}$ is inversible for all values of λ, and one can always guarantee a finite error bound with respect to the best projection (see Schoknecht (2002); Yu and Bertsekas (2008); Scherrer (2010)). If the goal of BRM(λ) is to compute $\bar{A}$ and $\bar{b}$ from samples, what it actually computes ($\tilde{A}$ and $\tilde{b}$) will in general be biased because it is based on a single trajectory[7]. Such a bias adds an uncontrolled variance term to $\bar{A}$ and $\bar{b}$ (for instance, see Antos *et al.* (2006)) of which an interesting consequence is that $\tilde{A}$ remains non singular[8]. More precisely, there are two sources of bias in the estimation: one results from the non Monte-carlo evaluation (the fact that $\lambda < 1$) and the other from the use of the correlated importance sampling factors (as soon as one considers off-policy learning). The interested reader may check that in the on-policy case, and when λ tends to 1, $\tilde{A}$ and $\tilde{b}$ coincide with $\bar{A}$ and $\bar{b}$. However, in the strictly off-policy case, taking $\lambda = 1$ does not prevent the bias due to the correlated importance sampling factors. If we have argued that LSTD/LSPE/FPKF asymptotically coincide when $\lambda = 1$, we see here that BRM will generally differ in an off-policy situation.

4. Illustrations of the algorithms

In this section, we briefly illustrate the behavior of all the algorithms we have described so far. In a first set of experiments, we consider random Markov chains involving 3 states and 2 actions and projections onto random spaces[9] of dimension 2. The discount factor is $\gamma = 0.99$. For each experiment, we have run all algorithms (plus TD(λ) with stepsize $\alpha_t = \frac{1}{t+1}$) 50 times with initial matrix (M_0, N_0, C_0) equal to[10] $100I$, with $\theta_0 = 0$ and during 100 000 iterations. For each of these 50 runs, the different algorithms share the same samples, that are generated by a random uniform policy π_0 (*i.e.*, that chooses each action with probability 0.5). We consider two situations: *on-policy*, where the policy to evaluate is $\pi = \pi_0$, and *off-policy*, where the policy to evaluate is random (*i.e.*, it picks the actions with probabilities p and $1 - p$, where p is chosen uniformly at random). In the curves we are about to describe, we display on the abscissa the iteration number and on the ordinate the median value of the distance (quadratic, weighted by the stationary distribution of P) between the computed value $\Phi\theta$ and the real value $V = (I - \gamma P)^{-1}R$ (*i.e.*, the lower the better).

For each of the two situations (*on-* and *off-policy*), we present data in two ways. To appreciate the influence of λ, we display the curves on one graph per algorithm with different values of λ (Fig. 1 and 2). To compare the algorithms for solving the Bellman equation $V = T^\lambda V$, we show on one graph per value of λ the error for the different algorithms (Fig. 3 and 4). From these experiments, we make the following observations:

7. It is possible to remove the bias when $\lambda = 0$ by using double samples. However, in the case where $\lambda > 0$, the possibility to remove the bias seems much more difficult: the natural solution involves generating an infinite number of trajectories.
8. $\bar{A}$ is by construction positive definite, and $\tilde{A}$ equals $\bar{A}$ plus a positive term (the variance term), and is thus also positive definite.
9. For each action, rewards are uniform random vectors on $(0, 1)^3$, transition matrices are random uniform matrices on $(0, 1)^{3\times 3}$ normalized so that the probabilities sum to 1. Random projections are induced by random uniform matrices Φ of size 3×2.
10. This matrix acts as an L_2 regularization and is used to avoid numerical instabilities at the beginning of the algorithms. The bigger the value, the smaller the influence.

- **In the on-policy setting,** LSTD and LSPE have similar performance and convergence speed for all values of λ. They tend to converge much faster than FPKF, which is slightly faster than TD. BRM is usually in between LSTD/LSPE and FPKF/TD, though for small values of λ, the bias seems significative. When λ increases, the performance of FPKF and BRM improves. At the limit when $\lambda = 1$, all algorithms (except TD) coincide (confirming the intuition for $\lambda = 1$, the influence of the choice ξ vanishes in Equation (3)).

- **In the off-policy setting,** LSTD and LSPE still share the same behavior. The drawbacks of the other algorithms are amplified with respect to the on-line situation. As λ increases, the performance of FPKF catches that of LSTD/LSPE. However, the performance of BRM seems to worsen while λ is increased from 0 to 0.99 and eventually approaches that of the other algorithms when $\lambda = 1$ (though it remains different, *cf.* the discussion in the previous section).

Globally, the use of eligibility traces allows to significantly improve the performance of FPKF(λ) over FPKF of Choi and Van Roy (2006) in both on- and off-policy cases, and that of BRM(λ) over BRM/GPTD/KTD of Engel (2005); Geist and Pietquin (2010b) in the on-policy case. The performance of BRM(λ) in the off-policy case is a bit disappointing, probably because of its inherent bias, which deserves further investigation. However, LSTD(λ)/LSPE(λ) appear to be in general the best algorithms.

Eventually, we consider two experiments involving an MDP and a projection due to Tsitsiklis and Van Roy (1997), in order to illustrate possible numerical issues when solving the projected fixed-point Equation (7). In the first experiment one sets (λ, γ) such that $\Pi_0 T^\lambda$ is expansive; as expected one sees (Fig. 5) that LSPE and FPKF both diverge. In the latter experiment, one sets (λ, γ) so that the spectral radius of $\Pi_0 T^\lambda$ is 1 (so that A is singular), and in this case LSTD also diverges (Fig. 6). In both situations, BRM is the only one not to diverge[11].

5. Conclusion

We have considered least-squares algorithms for value estimation in an MDP context. Starting from the on-policy case with no trace, we have recalled that several algorithms (LSTD, LSPE, FPKF and BRM/GPTD/KTD) optimize similar cost functions. Substituting the original Bellman operator by an operator that deals with traces and off-policy samples leads to the state-of-the-art off-policy trace-based versions of LSTD and LSPE, and suggests natural extensions of FPKF and BRM.

We have described recursive implementations of these algorithms and discussed their convergence properties. In particular, we have provided an original convergence analysis of BRM(λ) for sufficiently small λ, that implies the (so far not known) convergence of GPTD/KTD. Eventually, we have illustrated the behavior of these algorithms; to our knowledge, this constitutes the first comparison involving all these least-squares algorithms.

11. Note that this adverserial setting is meant to illustrate the fact that for the problem considered, some values (λ, γ) may be problematic for LSTD/LSPE/FPKF. In practice, λ can be chosen big enough so that these algorithms will be stable.

Overall, our study suggests that even if the use of eligibility traces generally improves the efficiency of the algorithms, LSTD(λ) and LSPE(λ) remain in general better than FPKF(λ) (that is much slower) and BRM(λ) (that may suffer from high bias). Furthermore, since LSPE(λ) requires more conditions for stability, LSTD(λ) probably remains the best choice in practice.

References

Antos, A., Szepesvári, C., and Munos, R. (2006). Learning Near-optimal Policies with Bellman-residual Minimization based Fitted Policy Iteration and a Single Sample Path. In *COLT*.

Baird, L. C. (1995). Residual Algorithms: Reinforcement Learning with Function Approximation. In *ICML*.

Bertsekas, D. and Ioffe, S. (1996). Temporal differences-based policy iteration and applications i n neuro-dynamic programming. Technical report, MIT.

Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific.

Bertsekas, D. P. and Yu, H. (2009). Projected Equation Methods for Approximate Solution of Large Linear Systems. *J. Comp. and Applied Mathematics*, **227**(1), 27–50.

Boyan, J. A. (1999). Technical Update: Least-Squares Temporal Difference Learning. *Machine Learning*, **49**(2-3), 233–246.

Bradtke, S. J. and Barto, A. G. (1996). Linear Least-Squares algorithms for temporal difference learning. *Machine Learning*, **22**(1-3), 33–57.

Choi, D. and Van Roy, B. (2006). A Generalized Kalman Filter for Fixed Point Approximation and Efficient Temporal-Difference Learning. *DEDS*, **16**, 207–239.

Engel, Y. (2005). *Algorithms and Representations for Reinforcement Learning*. Ph.D. thesis, Hebrew University.

Engel, Y., Mannor, S., and Meir, R. (2005). Reinforcement Learning with Gaussian Processes. In *ICML*.

Geist, M. and Pietquin, O. (2010a). Eligibility Traces through Colored Noises. In *ICUMT*.

Geist, M. and Pietquin, O. (2010b). Kalman Temporal Differences. *JAIR*, **39**, 483–532.

Geist, M. and Pietquin, O. (2011). Parametric Value Function Approximation: a Unified View. In *ADPRL*.

Kearns, M. and Singh, S. (2000). Bias-Variance Error Bounds for Temporal Difference Updates. In *COLT*.

Maei, H. R. and Sutton, R. S. (2010). GQ(λ): A general gradient algorithm for temporal-difference prediction learning with eligibility traces. In *Conference on Artificial General Intelligence*.

Munos, R. (2003). Error Bounds for Approximate Policy Iteration. In *ICML*.

Nedić, A. and Bertsekas, D. P. (2003). Least Squares Policy Evaluation Algorithms with Linear Function Approximation. *DEDS*, **13**, 79–110.

Precup, D., Sutton, R. S., and Singh, S. P. (2000). Eligibility Traces for Off-Policy Policy Evaluation. In *ICML*.

Ripley, B. D. (1987). *Stochastic Simulation*. Wiley & Sons.

Scherrer, B. (2010). Should one compute the Temporal Difference fix point or minimize the Bellman Residual? The unified oblique projection view. In *ICML*.

Schoknecht, R. (2002). Optimality of Reinforcement Learning Algorithms with Linear Function Approximation. In *NIPS*.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. MIT Press, 3rd edition.

Tsitsiklis, J. and Van Roy, B. (1997). An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, **42**(5), 674–690.

Yu, H. (2010). Convergence of Least-Squares Temporal Difference Methods under General Conditions. In *ICML*.

Yu, H. and Bertsekas, D. (2008). New Error Bounds for Approximations from Projected Linear Equations. Technical Report C-2008-43, Dept. Computer Science, Univ. of Helsinki.

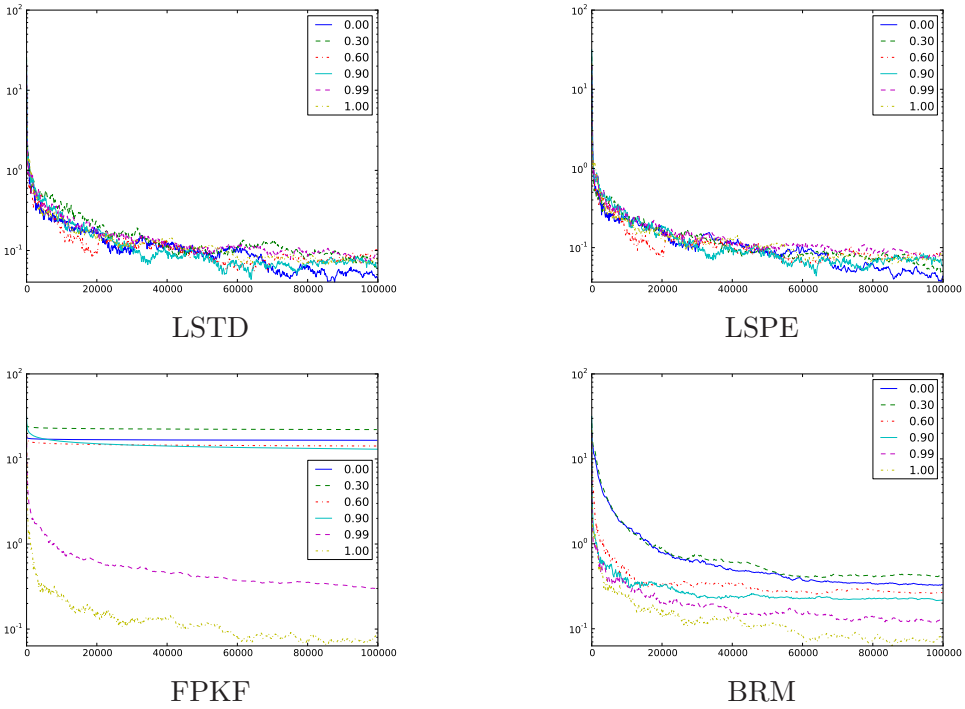


Figure 1: Influence of λ, *on-policy*

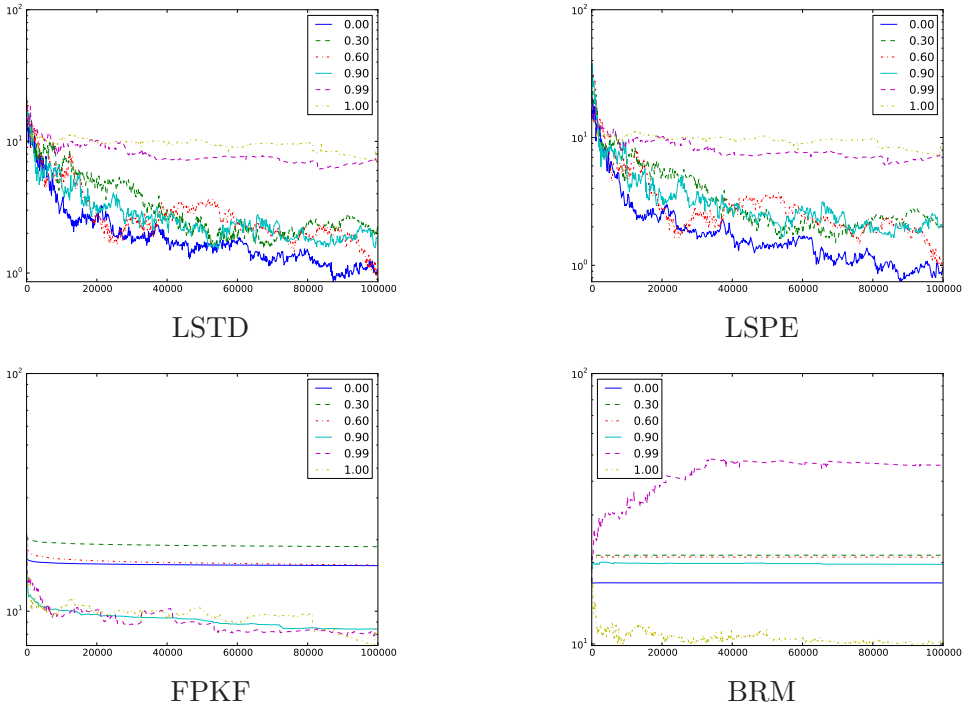
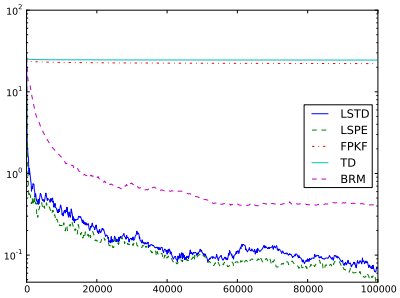
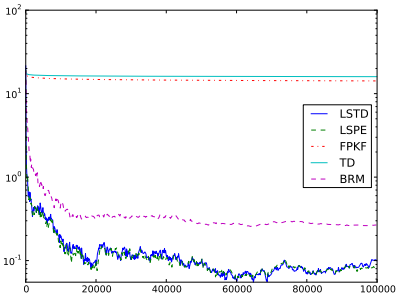


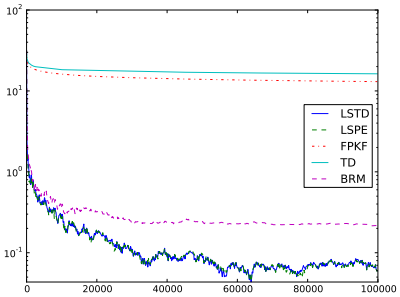
Figure 2: Influence of λ, *off-policy*



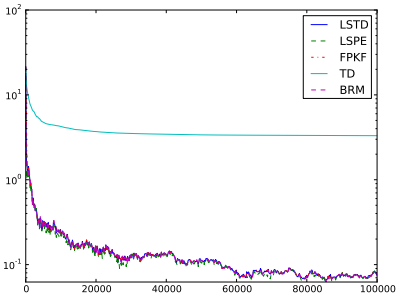
$\lambda = 0.3$



$\lambda = 0.6$

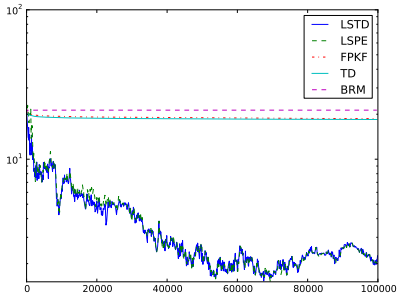


$\lambda = 0.9$

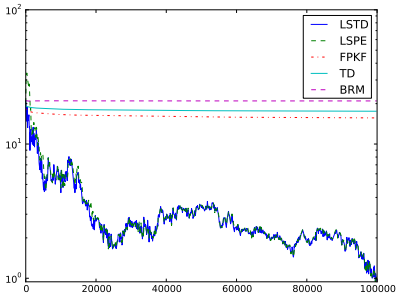


$\lambda = 1$

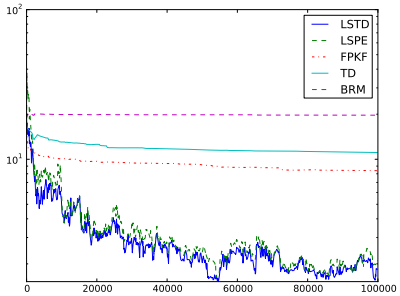
Figure 3: Comparison of the algorithms, *on-policy*



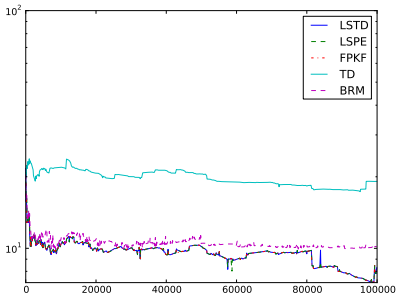
$\lambda = 0.3$



$\lambda = 0.6$



$\lambda = 0.9$



$\lambda = 1$

Figure 4: Comparison of the algorithms, *off-policy*

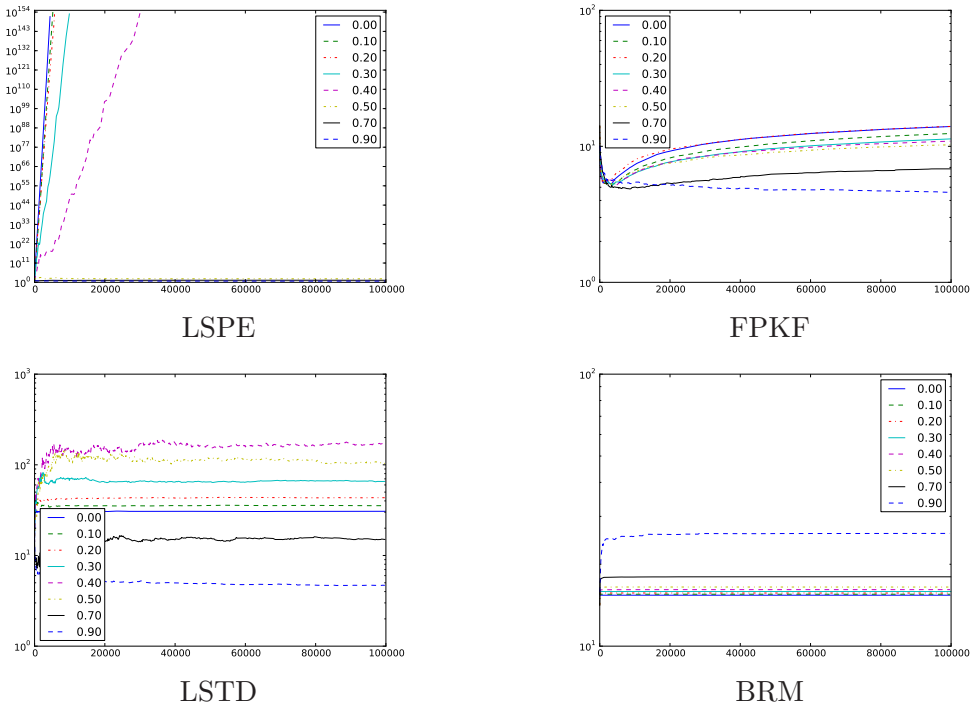


Figure 5: Pathological situation where LSPE and FPKF diverge (while LSTD converges)

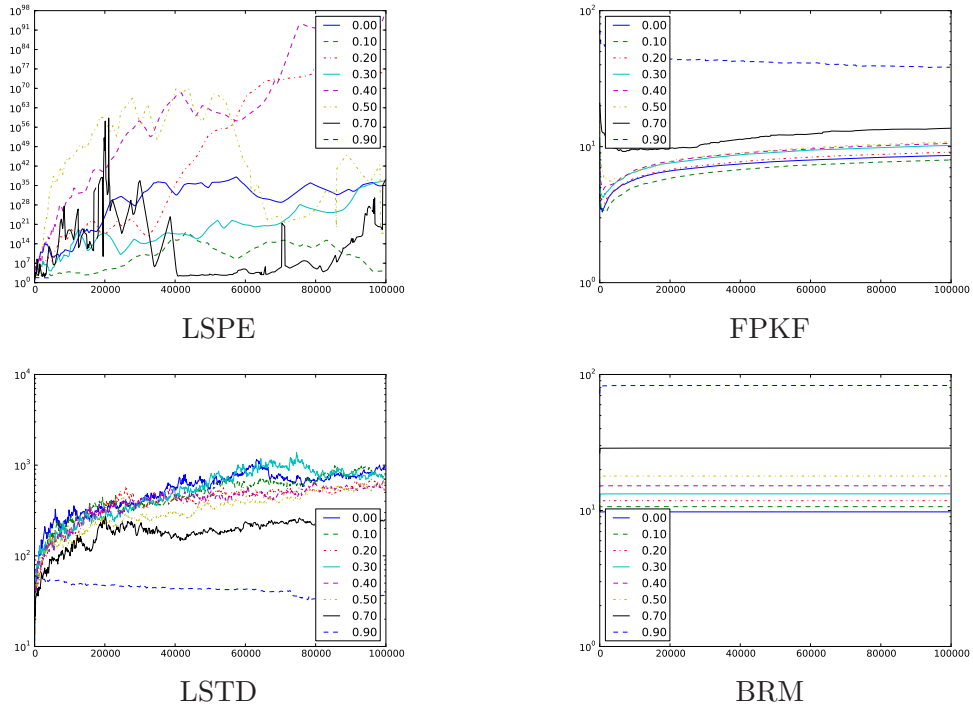


Figure 6: Pathological situation where LSPE, FPKF and LSTD all diverge.

Appendix: Proof of Theorem 5 (Convergence of BRM(λ))

The proof of Theorem 5 follows the general idea of that of Proposition 4 of Bertsekas and Yu (2009). It is done in 2 steps. First we argue that the limit of the sequence is linked to that of an alternative algorithm for which one cuts the traces at a certain depth l. Then, we show that for all depth l, this alternative algorithm converges almost surely, we explicitely compute its limit and make l tend to infinity to obtain the limit of BRM(λ).

We will only show that $\frac{1}{i}\tilde{A}_i$ tends to $\tilde{A}$. The argument is similar for $\frac{1}{i}b_i \to \tilde{b}$. Consider the following l-truncated version of the algorithm based on the following alternative traces (we here limit the "memory" of the traces to a size l):

$$y_{k,l} = \sum_{m=\max(1,k-l+1)}^{k} (\tilde{\rho}_m^{k-1})^2$$

$$\Delta_{j,l} = \sum_{k=\max(1,j-l+1)}^{j} \tilde{\rho}_k^{j-1} y_{k,l} \Delta\phi_k$$

and update the following matrix:

$$\tilde{A}_{i,l} = \tilde{A}_{i-1,l} + \Delta\phi_i \Delta\phi_i^T y_{i,l} + \tilde{\rho}_{i-1}(\Delta\phi_i \Delta_{i-1,l}^T + \Delta_{i-1,l}\Delta\phi_i^T).$$

The assumption in Equation (12) implies that $\tilde{\rho}_i^{j-1} \leq \beta^{j-i}$, therefore it can be seen that for all k,

$$|y_{k,l} - y_k| = \sum_{m=1}^{\max(0,k-l)} (\tilde{\rho}_m^{k-1})^2 \leq \sum_{m=1}^{\max(0,k-l)} \beta^{2(k-m)} \leq \frac{\beta^{2l}}{1-\beta^2} = \epsilon_1(l)$$

where $\epsilon_1(l)$ tends to 0 when l tends to infinity. Similarly, using the fact that $y_k \leq \frac{1}{1-\beta^2}$ and writing $K = \max_{s,s'} \|\phi(s) - \gamma\phi(s')\|_\infty$, one has for all j,

$$\begin{aligned}
\|\Delta_{j,l} - \Delta_j\|_\infty &\leq \sum_{k=1}^{\max(0,j-l)} \tilde{\rho}_k^{j-1} \|y_k \Delta\phi_k\|_\infty + \sum_{k=\max(1,j-l+1)}^{j} \tilde{\rho}_k^{j-1} |y_{k,l} - y_k| \|\Delta\phi_k\|_\infty \\
&\leq \sum_{k=1}^{\max(0,j-l)} \tilde{\rho}_k^{j-1} \frac{1}{1-\beta^2} K + \sum_{k=\max(1,j-l+1)}^{j} \tilde{\rho}_k^{j-1} \frac{\beta^{2l}}{1-\beta^2} K \\
&\leq \frac{\beta^l}{1-\beta} \frac{1}{1-\beta^2} K + \frac{1}{1-\beta} \frac{\beta^{2l}}{1-\beta^2} K = \epsilon_2(l)
\end{aligned}$$

where $\epsilon_2(l)$ also tends to 0. Then, it can be seen that:

$$\begin{aligned}
\|\tilde{A}_{i,l} - \tilde{A}_i\|_\infty &= \Big\| \tilde{A}_{i-1,l} - \tilde{A}_{i-1} + \Delta\phi_i \Delta\phi_i^T (y_{i,l} - y_i) \\
&\qquad + \tilde{\rho}_{i-1}(\Delta\phi_i(\Delta_{i-1,l}^T - \Delta_{i-1}^T) + (\Delta_{i-1,l} - \Delta_{i-1})\Delta\phi_i^T) \Big\|_\infty \\
&\leq \|\tilde{A}_{i-1,l} - \tilde{A}_{i-1}\|_\infty + \|\Delta\phi_i \Delta\phi_i^T\|_\infty |y_{k,l} - y_k| + 2\beta \|\Delta\phi_i\|_\infty \|\Delta_{i-1,l} - \Delta_i\|_\infty \\
&\leq \|\tilde{A}_{i-1,l} - \tilde{A}_{i-1}\|_\infty + K^2 \epsilon_1(l) + 2\beta K \epsilon_2(l)
\end{aligned}$$

and, by a recurrence on i, one obtains

$$\left\| \frac{\tilde{A}_{i,l}}{i} - \frac{\tilde{A}_i}{i} \right\|_\infty \leq \epsilon(l)$$

where $\epsilon(l)$ tends to 0 when l tends to infinity. This implies that:

$$\liminf_{l\to\infty} \frac{\tilde{A}_{i,l}}{i} - \epsilon(l) \leq \liminf_{l\to\infty} \frac{\tilde{A}_i}{i} \leq \limsup_{l\to\infty} \frac{\tilde{A}_i}{i} \leq \limsup_{l\to\infty} \frac{\tilde{A}_{i,l}}{i} + \epsilon(l).$$

In other words, one can see that $\lim_{i\to\infty} \frac{\tilde{A}_i}{i}$ and $\lim_{l\to\infty} \lim_{i\to\infty} \frac{\tilde{A}_{i,l}}{i}$ are equal if the latter exists. In the remaing of the proof, we show that the latter limit indeed exists and we compute it explicitely.

Let us fix some l and let us consider the sequence $(\frac{\tilde{A}_{i,l}}{i})$. At some index i, $y_{i,l}$ depends only on the last l samples, while $\Delta_{i,l}$ depends on the same samples and the last l values of $y_{j,l}$, thus on the last $2l$ samples. It is then natural to view the computation of $\tilde{A}_{i,l}$, which is based on $y_{i,l}$, $\Delta_{i-1,l}$ and $\Delta\phi_i = \phi_i - \gamma\rho_i\phi_{i+1}$, as being related to a Markov chain of which the states are the $2l+1$ consecutive states of the original chain $(s_{i-2l}, \ldots, s_i, s_{i+1})$. Write E_0 the expectation with respect to its stationary distribution. By the Markov chain Ergodic Theorem, we have with probability 1:

$$\lim_{i\to\infty} \frac{\tilde{A}_{i,l}}{i} = E_0\left[\Delta\phi_{2l}\Delta\phi_{2l}^T y_{2l,l} + \lambda\gamma\rho_{2l-1}(\Delta\phi_{2l}\Delta_{2l-1,l}^T + \Delta_{2l-1,l}\Delta\phi_{2l}^T)\right]. \tag{13}$$

Let us now explicitely compute this expectation. Write x_i the indicator vector (of which the k^{th} coordinate equals 1 when the state at time i is k and 0 otherwise). One has the following relations: $\phi_i = \Phi^T x_i$. Let us first look at the left part of the above limit:

$$\begin{aligned}
E_0\left[\Delta\phi_{2l}\Delta\phi_{2l}^T y_{2l,l}\right] &= E_0\left[(\phi_{2l} - \gamma\rho_{2l}\phi_{2l+1})(\phi_{2l} - \gamma\rho_{2l}\phi_{2l+1})^T y_{2l,l}\right] \\
&= E_0\left[\Phi^T(x_{2l} - \gamma\rho_{2l}x_{2l+1})(x_{2l} - \gamma\rho_{2l}x_{2l+1})^T\Phi\left(\sum_{m=l+1}^{2l}(\lambda\gamma)^{2(2l-m)}(\rho_m^{2l-1})^2\right)\right] \\
&= \Phi^T\left\{\sum_{m=l+1}^{2l}(\lambda\gamma)^{2(2l-m)}E_0\left[(\rho_m^{2l-1})^2(x_{2l} - \gamma\rho_{2l}x_{2l+1})(x_{2l} - \gamma\rho_{2l}x_{2l+1})^T\right]\right\}\Phi \\
&= \Phi^T\left\{\sum_{m=l+1}^{2l}(\lambda\gamma)^{2(2l-m)}E_0\left[(X_{m,2l,2l} - \gamma X_{m,2l,2l+1} - \gamma X_{m,2l+1,2l} + \gamma^2 X_{m,2l+1,2l+1})\right]\right\}\Phi
\end{aligned}$$

where we used the definiton $\tilde{\rho}_j^{k-1} = (\lambda\gamma)^{k-j}\rho_j^{k-1}$ and the notation $X_{m,i,j} = \rho_m^{i-1}\rho_m^{j-1}x_i x_j^T$. To finish the computation, we will mainly rely on the following Lemma:

Lemma 6 (Some identities) *Let $\tilde{P}$ be the matrix of which the coordinates are $\tilde{p}_{ss'} = \sum_a \pi(s,a)\rho(s,a)T(s,a,s')$, which is in general not a stochastic matrix. Let μ_0 be the stationary distribution of the behavior policy π_0. Write $\tilde{D}_i = \mathrm{diag}\left((\tilde{P}^T)^i\mu_0\right)$. Then*

$$\begin{aligned}
\forall m \leq i,\ E_0[X_{m,i,i}] &= \tilde{D}_{i-m} \\
\forall m \leq i \leq j,\ E_0[X_{m,i,j}] &= \tilde{D}_{i-m}P^{j-i} \\
\forall m \leq j \leq i,\ E_0[X_{m,i,j}] &= (P^T)^{j-i}\tilde{D}_{i-m}
\end{aligned}$$

Proof We first observe that:

$$\begin{aligned} E_0[X_{m,i,i}] &= E_0[(\rho_m^{i-1})^2 x_i x_i^T] \\ &= E_0[(\rho_m^{i-1})^2 \operatorname{diag}(x_i)] \\ &= \operatorname{diag}\left(E_0[(\rho_m^{i-1})^2 x_i\right) \end{aligned}$$

To provide the identity, we will thus simply provide a proof by recurrence that $E_0[(\rho_m^{i-1})^2 x_i] = (\tilde{P}^T)^{m-i}\mu_0$. For $i = m$, we have $E_0[x_m] = \mu_0$. Now suppose the relation holds for i and let us prove it for $i+1$.

$$\begin{aligned} E_0[(\rho_m^i)^2 x_{i+1}] &= E_0\left[E_0[(\rho_m^i)^2 x_{i+1}|\mathcal{F}_i]\right] \\ &= E_0\left[E_0[(\rho_m^{i-1})^2(\rho_i)^2 x_{i+1}|\mathcal{F}_i]\right] \\ &= E_0\left[(\rho_m^{i-1})^2 E_0[(\rho_i)^2 x_{i+1}|\mathcal{F}_i]\right]. \end{aligned}$$

Write $\mathcal{F}_i$ the realization of the process until time i. Recalling that s_i is the state at time i and x_i is the indicator vector corresponding to s_i, one has for all s':

$$\begin{aligned} E_0[(\rho_i)^2 x_{i+1}(s')|\mathcal{F}_i] &= \sum_a \pi_0(s_i,a)\rho(s_i,a)^2 T(s_i,a,s') \\ &= \sum_a \pi(s_i,a)\rho(s_i,a) T(s_i,a,s') \\ &= \tilde{p}_{s_i,s'} \\ &= [\tilde{P}^T x_i](s'). \end{aligned}$$

As this is true for all s', we deduce that $E_0[(\rho_i)^2 x_{i+1}|\mathcal{F}_i] = \tilde{P}^T x_i$ and

$$\begin{aligned} E_0[(\rho_m^i)^2 x_{i+1}] &= E_0[(\rho_m^{i-1})^2 \tilde{P}^T x_i] \\ &= \tilde{P}^T E_0[(\rho_m^{i-1})^2 \tilde{P}^T x_i] \\ &= \tilde{P}^T (\tilde{P}^T)^i \mu_0 \\ &= (\tilde{P}^T)^{i+1}\mu_0 \end{aligned}$$

which concludes the proof by recurrence.

Let us consider the next identity. For $i \le j$,

$$\begin{aligned} E_0[\rho_m^{i-1}\rho_m^{j-1} x_i x_j^T] &= E_0[E_0[\rho_m^{i-1}\rho_m^{j-1} x_i x_j^T|\mathcal{F}_i]] \\ &= E_0[(\rho_m^{i-1})^2 x_i E_0[\rho_i^{j-1} x_j^T|\mathcal{F}_i]] \\ &= E_0[(\rho_m^{i-1})^2 x_i x_i^T P^{j-i}] \\ &= \operatorname{diag}\left((\tilde{P}^T)^{m-i}\mu_0\right) P^{j-i}. \end{aligned}$$

Eventually, the last identity is obtained by considering $Y_{m,i,j} = X_{m,j,i}^T$. ■

Thus, coming back to our calculus,

$$E_0\left[\Delta\phi_{2l}\Delta\phi_{2l}^T y_{2l,l}\right] = \Phi^T\left\{\sum_{m=l+1}^{2l}(\lambda\gamma)^{2(2l-m)}\left(\tilde{D}_{2l-m} - \gamma\tilde{D}_{2l-m}P - \gamma P^T\tilde{D}_{2l-m} + \gamma^2\tilde{D}_{2l+1-m}\right)\right\}\Phi$$

$$= \Phi^T(D_l - \gamma D_l P - \gamma P^T D_l + \gamma^2 D'_l)\Phi \tag{14}$$

$$\text{with} \quad D_l = \sum_{j=0}^{l-1}(\lambda\gamma)^{2j}\tilde{D}_j, \quad \text{and} \quad D'_l = \sum_{j=0}^{l-1}(\lambda\gamma)^{2j}\tilde{D}_{j+1}.$$

Similarly, the second term on the right side of Equation (13) satisfies:

$$E_0\left[\rho_{2l-1}\Delta_{2l-1,l}\Delta\phi_{2l}^T\right] = E_0\left[\rho_{2l-1}\sum_{k=l}^{2l-1}\tilde{\rho}_k^{2l-2}y_{k,l}\Delta\phi_k\Delta\phi_{2l}^T\right]$$

$$= E_0\left[\sum_{k=l}^{2l-1}(\lambda\gamma)^{2l-1-k}\rho_k^{2l-1}\left(\sum_{m=k-l+1}^{k}(\tilde{\rho}_m^{k-1})^2\right)\Phi^T(x_k - \gamma\rho_k x_{k+1})(x_{2l} - \gamma\rho_{2l}x_{2l+1})^T\Phi\Delta\phi_{2l}^T\right]$$

$$= \Phi^T\left(\sum_{k=l}^{2l-1}(\lambda\gamma)^{2l-1-k}\sum_{m=k-l+1}^{k}(\lambda\gamma)^{2(k-m)}E_0\left[\rho_m^{2l-1}\rho_m^{k-1}(x_k - \gamma\rho_k x_{k+1})(x_{2l} - \gamma\rho_{2l}x_{2l+1})^T\right]\right)\Phi$$

$$= \Phi^T\left(\sum_{k=l}^{2l-1}(\lambda\gamma)^{2l-1-k}\sum_{m=k-l+1}^{k}(\lambda\gamma)^{2(k-m)}E_0\left[X_{m,k,2l} - \gamma X_{m,k+1,2l} - \gamma X_{m,k,2l+1} + \gamma^2 X_{m,k+1,2l+1}\right]\right)\Phi$$

$$= \Phi^T\left(\sum_{k=l}^{2l-1}(\lambda\gamma)^{2l-1-k}\sum_{m=k-l+1}^{k}(\lambda\gamma)^{2(k-m)}\left(\tilde{D}_{k-m}P^{2l-k} - \gamma\tilde{D}_{k+1-m}P^{2l-k-1} - \gamma\tilde{D}_{k-m}P^{2l+1-k} + \gamma^2\tilde{D}_{k+1-m}P^{2l-k}\right)\right)\Phi$$

$$= \Phi^T\left(\sum_{k=l}^{2l-1}(\lambda\gamma)^{2l-1-k}\sum_{m=k-l+1}^{k}(\lambda\gamma)^{2(k-m)}\left(\tilde{D}_{k-m}P^{2l-k}(I-\gamma P) - \gamma\tilde{D}_{k+1-m}P^{2l-1-k}(I-\gamma P)\right)\right)\Phi$$

$$= \Phi^T\left(\sum_{k=l}^{2l-1}(\lambda\gamma)^{2l-1-k}\sum_{m=k-l+1}^{k}(\lambda\gamma)^{2(k-m)}\left(\tilde{D}_{k-m}P - \gamma\tilde{D}_{k+1-m}\right)P^{2l-1-k}(I-\gamma P)\right)\Phi$$

$$= \Phi^T\left(\sum_{k=l}^{2l-1}(\lambda\gamma)^{2l-1-k}\left(D_l P - \gamma D'_l\right)P^{2l-1-k}(I-\gamma P)\right)\Phi$$

$$= \Phi^T\left(D_l P - \gamma D'_l\right)Q_l(I-\gamma P)\Phi$$

with $Q_l = \sum_{j=0}^{l-1}(\lambda\gamma P)^j$.

Gathering this and Equation (14), we see that the limit of $\frac{A_{i,l}}{i}$ expressed in Equation (13) equals:

$$\Phi^T\left[D_l - \gamma D_l P - \gamma P^T D_l + \gamma^2 D'_l + \lambda\gamma\left((D_l P - \gamma D'_l)Q_l(I-\gamma P) + (I-\gamma P^T)Q_l^T(P^T D_l - \gamma D'_l)\right)\right]\Phi.$$

When l tends to infinity, Q_l tends to $Q = (I-\lambda\gamma P)^{-1}$. The assumption of Equation (12) ensures that $(\lambda\gamma)\tilde{P}$ has spectral radius smaller than 1, and thus when l tends to infinity, D_l tends to $D = \text{diag}\left((I-(\lambda\gamma)^2\tilde{P}^T)^{-1}\mu_0\right)$ and D'_l to $D' = \text{diag}\left(\tilde{P}^T(I-(\lambda\gamma)^2\tilde{P}^T)^{-1}\mu_0\right)$.

In other words, $\lim_{l\to\infty}\lim_{i\to\infty}\frac{\tilde{A}_{i,l}}{i}$ exists with probability 1 and equals:

$$\Phi^T\left[D-\gamma DP-\gamma P^TD+\gamma^2D'+\lambda\gamma\left((DP-\gamma D')Q(I-\gamma P)+(I-\gamma P^T)Q^T(P^TD-\gamma D')\right)\right]\Phi.$$

Eventually, this shows that $\lim_{i\to\infty}\frac{\tilde{A}_i}{i}$ exists with probability 1 and shares the same value.

A similar reasoning allows to show that $\lim_{i\to\infty}\frac{\tilde{b}_i}{i}$ exists and equals

$$\Phi^T\left[(I-\gamma P^T)Q^TD+\lambda\gamma(DP-\gamma D')Q\right]R^\pi.\blacksquare$$