Moindres carrés récursifs pour l'évaluation off-policy d'une politique avec traces d'éligibilité

Abstract : Dans le cadre des processus de décision Markoviens (MDPs), nous nous intéressons à l'apprentissage d'une *approximation linéaire* de la fonction de valeur d'une politique fixe, lorsque les données sont constituées d'une unique trajectoire générée par une autre politique, c'est-à-dire que nous considérons le cas *off-policy*. Nous présentons une manière systématique d'adapter l'ensemble des algorithmes de type moindres carrés proposés dans la littérature dans le cas *on-policy* et n'utilisant pas nécessairement de traces d'éligibilité (LSTD (Boyan, 1999), LSPE) (Nedić & Bertsekas, 2003), FPKF (Choi & Roy, 2006) et BRM/GPTD (Engel, 2005)/KTD (Geist & Pietquin, 2010c)), de sorte à ce qu'ils puissent être appliqués dans le cas *off-policy* avec des traces. Nous dérivons les formules pour une implémentation récursive de ces algorithmes, étudions leur convergence asymptotique et illustrons expérimentalement leur comportement. Si nous retrouvons les algorithmes off-policy LSTD(λ)/LSPE(λ) récemment proposés par Yu (2010), les deux autres sont à notre connaissance nouveaux.

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. We focus on the situation where learning is done from a single trajectory possibly generated by some other policy, which is also known as *off-policy* learning.

Given samples, simple methods for estimating a value function are temporal difference (TD) learning (Sutton & Barto, 1998) and Monte Carlo. TD learning with eligibility traces (Sutton & Barto, 1998), known as TD(λ), provide a nice bridge between both TD and Monte-Carlo, and by controlling the bias/variance trade-off (Kearns & 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 & Van Roy, 1997). Overall, the use of these traces (and the setting of λ) often plays an important practical role.

Only a decade ago did Precup *et al.* (2000) propose the first variation of TD(λ) that could combine off-policy learning with linear approximation and eligibility traces. Much more recently, Yu (2010) proposed off-policy LSTD(λ)/LSPE(λ), Least-Squares (LS) algorithms – LS algorithms are usually much more efficient in terms of samples than their stochastic approximation couterparts like TD – that also use eligibility traces.

On-policy learning (where the policy to evaluate is the same as the one that generated data) has a much longer history in the literature; thus several on-policy LS algorithms were proposed in the past, notably LSTD(λ) (Boyan, 1999), LSPE(λ) (Nedić & Bertsekas, 2003) that use eligibility traces, FPKF (Choi & Roy, 2006) and GPTD (Engel, 2005)/KTD (Geist & Pietquin, 2010c) that do not[1]. The first motivation of this article is to argue that it is conceptually simple to extend the just-mentioned algorithms so that they can be also applied to the off-policy setting, while keeping the use of eligibility traces. If this allows to rederive the off-policy LSTD(λ)/LSPE(λ) algorithms of Yu (2010), it also allows to define two new LS algorithms. The second motivation of this work is of algorithmic nature: we explicitely derive formulas that allow to run these new algorithms in a recursive manner, i.e. where each sample of the trajectory can be processed on-the-fly with a complexity quadratic in the number of features. To our knowledge, this has not even been done for LSPE(λ).

[1] GPTD has been extended to the case $\lambda = 1$ (Engel *et al.*, 2005) and KTD to $\lambda \in [0, 1]$ (Geist & Pietquin, 2010b). However, for KTD(λ), it is not really the same traces.

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 LS 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, briefly commented, and the formula for their recursive implementation is derived. Section 4 illustrates empirically the behavior of these algorithms and Section 5 concludes and describes future work.

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 $1, 2, \dots, N$, 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 deterministic reward function and γ the discount factor. The 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. 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 means that the expectation is trajectories induced by policy π. The value function satisfies the (linear) Bellman evaluation 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: $\hat{V}_\theta(s) = \theta^T \phi(s)$ with $\theta \in \mathbb{R}^p$ being the parameter vector and $\phi(s)$ the feature vector. We also want to estimate the value function V^π (or equivalently associated parameters) 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 asumme it exists and is unique). Let D_0 be the diagonal matrix of which the elements are $(\mu_0(s_i))_{1 \le i \le |S|}$. Let Φ be the matrix of feature vectors: $\Phi = [\phi(1) \dots \phi(N)]^T$. The projection Π_0 onto the space spanned by Φ with respect to the μ_0-quadratic norm has the following closed-form:

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

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 π_0 and π are the same so we omit the corresponding superscripts/subscripts. 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 target policy π is available. Let us introduce the sampled Bellman operator $\hat{T}_j$, defined as:

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

Notice that $\hat{T}_j V$ is an unbiased estimate of $TV(s_j)$. If values were observable, estimating the parameter vector θ would reduce to project the value function onto the hypothesis space using the empirical projection operator. This is the classical least-squares approach under the linear parameterization assumption. However, values are not observed, only rewards. Nevertheless, one can rely on temporal differences to estimate the value function.

The Least-Squares Temporal Differences (LSTD) algorithm of Bradtke & 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 $\hat{V}_\theta = \Pi_0 T \hat{V}_\theta$, Π_0 being the just introduced projection operator. Using the available trajectory, LSTD solves the following optimization 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 Nedić & Bertsekas (2003) searches for the same fixed point, but in an iterative way instead of directly (informally, $\hat{V}_{\theta_i} \simeq \Pi_0 T \hat{V}_{\theta_{i-1}}$). 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 & Roy (2006) is a least-squares variation of the classical temporal difference learning algorithm (Sutton & 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 (Choi & Roy, 2006, Sec. 1.6):

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

Finally, the Bellman Residual Minimization (BRM) algorithm aims at minimizing the distance between the value function and its image through the Bellman operator, $\|V - TV\|^2$. Not that when the sampled operator is used, this leads to biased estimates (see e.g. 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 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. The parametric Gaussian Process Temporal Differences (GPTD) algorithm of Engel (2005) and the linear Kalman Temporal Differences (KTD) algorithm of Geist & Pietquin (2010c) can be shown to minimize this cost using a least-squares approach (so with bias).

All these algorithms can be summarized as minimizing the following generic cost-function:

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

One 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. Actually, more algorithms can be summarized under this generic equation (Geist & Pietquin, 2010a), but this 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 correspond to looking for the fixed point of the following variation of the Bellman operator (Bertsekas & 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 e.g. Nedić & 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

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

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: $T^1V(s_i) = E_\pi[\sum_{j=i}^{\infty}\gamma^{j-1}r_j|s_i]$.

As learning is done over a finite trajectory, it is natural to introduce the following truncated operator T_n^λ:

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

To use it practically, we will need to remove the dependency to the model (i.e. the expectation). Moreover, the goal is now to learn the value function of the target policy π from a single trajectory sampled using a different policy.

Assume again that learning is done from some trajectory $(s_1, a_1, \ldots, s_j, a_j, r_j, s_{j+1}, \ldots s_{i+1})$, now sampled according to the known behaviour policy π_0. As behaviorial and target policies are different, it is not sufficient to remove the expectation to obtain an unbiased estimate of T_n^λ, one has to correct it using importance sampling (Ripley, 1987). For all s, a 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:

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

With these corrections, it can be seen that $\hat{T}_{i,n}^\lambda V$ is an unbiased estimate of $T_n^\lambda V(s_i)$ (the interested reader may consult (Precup *et al.*, 2000; Yu, 2010) for further 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 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 next subsection, by instantiating ξ to θ_i, θ_{i-1}, θ_{j-1} or ω, we derive the already existing algorithms off-policy LSTD(λ) and 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).

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:*

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

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:*

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

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 $\hat{T}_\pi^{\lambda,i}$ (forward view of eligibility traces) to the causal recursion over parameters (backward view of eligibility traces).

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 $\xi = \theta_i$:

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

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

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

$$\Leftrightarrow 0 = \sum_{j=1}^{i} \sum_{k=j}^{i} \phi_j \tilde{\rho}_j^{k-1} (\rho_k r_k - \Delta\phi_k^T \theta_i)$$

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 $\theta_0$ and matrix $M_0$ ;
Set $z_0 = 0$;
for $i = 1, 2, \dots$ do
    Observe $\phi_i, r_i, \phi_{i+1}$ ;
    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[2], 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:

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

[2] It is not always the case, see Tsitsiklis & Van Roy (1997) or Section 4 for counter examples.

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

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

where

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

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. Precisely, the author proves the following general result[3]:

Theorem 1 (Yu (2010))**.** *Assume that the stochastic matrix P_0 of the* behavior *policy is irreducible, and that for all states, any action that has a non-zero probability of being chosen by the* target *policy π also has a non-zero probability of being chosen by π_0 (formally: $\forall s, a,\ \pi(a|s) > 0 \Rightarrow \pi_0(a|s) > 0$). Then, for any function $\psi : S \times A \times S \to \mathbb{R}$,*

$$\frac{1}{i}\sum_{j=1}^{i} z_j \psi(s_j, a_j, s_{j+1}) \xrightarrow{i\to\infty,\ a.s.} \Phi^T D_0(I - \lambda\gamma P)^{-1}\Psi$$

where $(z_i)_{i\in\mathbb{N}}$ is the sequence of eligibility vectors defined in Equation (4), and the vector/matrix Ψ is defined in terms of its rows: $\Psi^T = (\bar{\psi}(1)^T \dots \bar{\psi}(n)^T)$ with $\bar{\psi}(i) = E[\psi(s_1, a_1, s_2)|s_1 = i, a_i \sim \pi_0(\dot{|}s_1)]$.

The convergence of $\frac{1}{i}A_i$ to A (resp. the convergence of $\frac{1}{i}b_i$ to b) is obtained for $\psi(s, a, s') = \phi(s) - \gamma\rho(s, a)\phi(s')$ (resp. $\psi(s, a, s') = \rho(s, a)r(s, a)$). 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 also the corrected eligibility vector z_j):

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

Define the matrix N_i as follows:

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

[3] This is an equivalent rewriting of Theorem 3.3 in Yu (2010).

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:

$$A_i = \sum_{j=1}^{i} z_j \Delta\phi_j^T = A_{i-1} + z_i \Delta\phi_{i+1}^T$$

$$b_i = \sum_{j=1}^{i} z_j \rho_j r_j = b_{i-1} + z_i \rho_i r_i$$

Therefore, the update of the parameter vector can be written as:

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

Given the recursive updates of N_i, A_i and b_i, this gives Alg. 2. This generalizes the LSPE(λ) algorithm of Nedić & Bertsekas (2003) to off-policy learning for any eligibility factor. Though briefly mentionned by Yu (2010), this is to our knowledge the first time that the LSPE(λ) is derived in an off-policy context along with recursive formula.

Algorithm 2: Off-policy LSPE(λ)

```
Initialization;
Initialize vector $\theta_0$ and matrix $N_0$ ;
Set $z_0 = 0$, $A_0 = 0$ and $b_0 = 0$;
for $i = 1, 2, \dots$ do
    Observe $\phi_i, r_i, \phi_{i+1}$;
    Update traces ;
    $z_i = \gamma\lambda\rho_{i-1}z_{i-1} + \phi_i$ ;
    Update parameters ;
    $N_i = N_{i-1} - \frac{N_{i-1}\phi_i\phi_i^T N_{i-1}}{1+\phi_i^T N_{i-1}\phi_i}$ ;
    $A_i = A_{i-1} + z_i\Delta\phi_i^T$;
    $b_i = b_{i-1} + \rho_i z_i r_i$;
    $\theta_i = \theta_{i-1} + N_i(b_i - A_i\theta_{i-1})$ ;
```

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 & 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))

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

The behavior of this sequence depends on whether the spectral radius of $\Pi_0 T_\pi^\lambda$ is smaller than 1 or not, in other words whether the sequence is contracting or not. Thus, the analyses of Yu (2010) and Nedić & 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 $\Pi_0 T_\lambda$ has spectral radius smaller than 1, 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 (independently) ensure such a desired contraction property. The first one is when one considers on-policy learning (see e.g. Nedić & Bertsekas

[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.

(2003), where the authors studied the on-policy case and uses this property in the proof). 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 with a sufficiently big value of the discount factor γ, a small value of λ can make $\Pi_0 T^\lambda$ expansive (see (Tsitsiklis & Van Roy, 1997) for $\lambda = 0$) and off-policy LSPE(λ) will then diverge.

3.4 Off-policy FPKF(λ)

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

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

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:

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

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 $1 \le k \le i$. Using the symmetry of the dot product $\Delta\phi_j^T \theta_{k-1} = \theta_{k-1}^T \Delta\phi_j$ (since they are vectors), it is still 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 Lemma 1 for N_i 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 parameters update as provided in Alg. 3. As LSPE(λ), this algorithm is fundamentally recursive. It generalizes the FPKF algorithm of Choi & Roy (2006) (originally only introduced without traces in the on-policy case) to eligibility traces as well as to off-policy learning. Due to its much more involved form (with 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. We however conjecture that off-policy FPKF(λ) has the same asymptotic behavior as LSPE(λ).

3.5 Off-policy BRM(λ)

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

Algorithm 3: Off-policy FPKF(λ)

```
Initialization;
Initialize vector $\theta_0$ and matrix $N_0$ ;
Set $z_0 = 0$ and $Z_0 = 0$;
for $i = 1, 2, \dots$ do
    Observe $\phi_i, r_i, \phi_{i+1}$;
    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 ;
    $N_i = N_{i-1} - \frac{N_{i-1}\phi_i\phi_i^T N_{i-1}}{1+\phi_i^T N_{i-1}\phi_i}$ ;
    $\theta_i = \theta_{i-1} + N_i(z_i\rho_i r_i - Z_i \Delta\phi_i)$ ;
```

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

Let us define $\psi_{j\to i}$ and $z_{j\to i}$ as:

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

Therefore, 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:

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

Let us write y_k as:

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

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:

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

Let us introduce z_j and Δ_j as:

$$
\begin{aligned}
z_j &= \sum_{k=1}^{j} \tilde{\rho}_k^{j-1} y_k \rho_k r_k = \gamma\lambda\rho_{j-1} z_{j-1} + \rho_j r_j y_j \\
\Delta_j &= \sum_{k=1}^{j} \tilde{\rho}_k^{j-1} y_k \Delta\phi_k = \gamma\lambda\rho_{j-1}\Delta_{j-1} + y_j \Delta\phi_j
\end{aligned}
$$

Using these notations, 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:

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

Let u_j and v_j denote:

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

We have that (I_2 denotes the 2×2 identity matrix):

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

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:

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

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

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

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.

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. Despite the closeness of GPTD, KTD and BRM, their extension to eligibility traces are different: GPTD(λ) (Engel, 2005), KTD(λ) (Geist &

Algorithm 4: Off-policy BRM(λ)

```
Initialization;
Initialize vector $\theta_0$ and matrix $C_0$ ;
Set $y_0 = 0$, $\Delta_0 = 0$ and $z_0 = 0$;
for $i = 1, 2, \dots$ do
    Observe $\phi_i, r_i, \phi_{i+1}$;
    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 ;
    $\theta_i = \theta_{i-1} + C_{i-1}U_i\left(I_2 + V_iC_{i-1}U_i\right)^{-1}(W_i - V_i\theta_{i-1})$ ;
    $C_i = C_{i-1} - C_{i-1}U_i\left(I_2 + V_iC_{i-1}U_i\right)^{-1}V_iC_{i-1}$ ;
    Post-update traces ;
    $\Delta_i = (\gamma\lambda\rho_{i-1})\Delta_{i-1} + \Delta\phi_i y_i$ ;
    $z_i = (\gamma\lambda\rho_{i-1})z_{i-1} + r_i\rho_i y_i$ ;
```

Pietquin, 2010b) and the provided BRM(λ) are different algorithms. Basically, GPTD(λ) mimics the LSTD(λ) algorithms and KTD(λ) uses a different Bellman operator[5].

We now provide some analysis for this new algorithm. We have a sufficient condition for proving the almost sure convergence of this algorithm. Without loss of generality (and in order to slightly simplify our result), we have restricted the analysis to the case where the reward function does not depend on the action, and thus can be written as a vector R of size n (in the algorithm all the terms $\rho_j r_j$ are then simply replaced by r_j).

Theorem 2. *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*

$$\forall (s,a), \quad \lambda\gamma\rho(s,a) \leq \beta, \tag{10}$$

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

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

where we wrote:

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

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 (10) 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. The matrix $\tilde{P}$, which is in general not a stochastic matrix, can have a spectral radius bigger than 1; Equation (10) ensures that $(\lambda\gamma)^2\tilde{P}$ has spectral radius smaller than β so that D and D' are well defined. Finally, note that there is probably no hope to completely remove such

[5] 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 $\hat{T}^1_{j,i}$

an assumption since by making $\lambda\gamma$ big enough, one may force the spectral radius of $(\lambda\gamma)^2\tilde{P}$ to be as close as one may want to 1, which would make $\tilde{A}$ and $\tilde{b}$ diverge.

The proof of this Theorem mimicks that of Proposition 4 in Bertsekas & Yu (2009) and is detailed in the Appendix. The overall arguments are the following: Equation (10) 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:

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

with $\Psi = \Phi - (1-\lambda)\gamma PQ\Phi$, and solve 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 & 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[6]. Such a bias adds an uncontrolled variance term to $\bar{A}$ and $\bar{b}$ (see Antos *et al.* (2006)), of which an interesting consequence is that $\tilde{A}$ stays inversible. More precisely, there are two sources of bias in the estimation: one results from the non Monte carlo evaluation (when $\lambda < 1$) and the other from the use of the same importance sampling factors (as soon as one considers off-policy). Indeed, the interested reader may check that in the on-policy case, and when λ tends to 1, the bias of $\tilde{A}$ and $\tilde{b}$ both tend to 0.

4 Experiments

In this section, we illustrate experimentally the behavior of all the algorithms we have described so far. In a first set of experiments, we have considered random Markov chains involving 3 states and 2 actions (for each action, rewards are uniform random vectors on $(0,1)^3$, transition probabilities are random uniform matrices on $(0,1)^{3\times 3}$) normalized so that the probabilities sum to 1) and projections onto random spaces of dimension 2 (induced by random uniform matrices Φ of size 3×2). 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[7] $100I$, 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 uniform. 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 the same data in two ways. To appreciate the influence of λ, we display the curves on one graph per algorithm with different values of λ (Figures 1 and 2). To compare the algorithms for solving the Bellman equation $V = T_\lambda V$, we

[6] 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.

[7] This matrix, which acts as an L_2 regularization, is used to avoid numerical instabilities at the beginning of the algorithms. The bigger the value, the smaller the influence.

show on one graph per value of λ the performance of the different algorithms (Figures 3 and 4). From these experiments, we make the following observations:

- 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 and 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 difference between all algorithms (except TD) vanish (for $\lambda = 1$ they all behave the same, which since 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, in particular the bias of BRM may be big even for rather big values of λ. When λ tends to 1, the performance of FPKF can catch that of LSTD/LSPE while BRM may still have a significative bias (this corresponds to situations where the assumption of Equation (10) of Theorem 2 does not hold).

Eventually, we have run two other sets of experiments, where we consider an MDP and a projection due to Tsitsiklis & Van Roy (1997), in order to show the numerical difficulties that may arise when solving the projected fixed point Equation (7). In the first experiment one sets $\lambda\gamma$ such that $\Pi_0 T^\lambda$ is expansive; as expected one sees (see Figure 5) that LSPE and FPKF both diverge. In the latter experiment, one sets $\lambda\gamma$ so that the spectral radius of $\Pi_0 T^\lambda$ is 1 (so that A is singular), and in this case LSTD also diverges (see Figure 6). In both situations, BRM gives better results.

5 Conclusion and future work

In this paper, we have considered Least Squares algorithms for approximating the value of some fixed policy in an MDP context. Starting from the on-policy case with no trace, we recalled that several algorithm (LSTD, LSPE, FPKF and BRM) optimize similar cost functions. By substituing the original Bellman operator by an operator that deals with traces and off-policy samples, one naturally rederive off-policy trace-based versions of LSTD and LSPE, and propose extensions of FPKF and BRM. With respect to the original algorithm FPKF proposed by Choi & Roy (2006) and BRM(0), the introduction of eligibility traces leads to a significant increase of performance.

To sum up, the first (conceptual) contribution of this paper is to provide a unified view on existing algorithms, which naturally leads to new algorithms. For all these algorithms, we derived recursive formulas so that they can process data on the fly and this constitutes our second (algorithmic) contribution. We have recalled the essence of the arguments proving for the almost sure convergence of LSTD(λ)/LSPE(λ), which are originally due to Yu (2010). We have provided an original analysis of BRM, which constitutes a third (theoretical) contribution. The analysis of FPKF(λ), which is still lacking, represents a natural interesting future work. The experiments we have run illustrate our analysis, notably the potential divergence of LSPE/FPKF/LSTD. In such situations, BRM, which is based on a well-defined problem, seems more reliable. Better controlling its inherent bias constitutes another (difficult) research direction.

References

Antos A., Szepesvári C. & Munos R. (2006). Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. In *COLT*. Published Online First: 14 Nov, 2007.

Baird L. C. (1995). Residual Algorithms: Reinforcement Learning with Function Approximation. In *Proceedings of the International Conference on Machine Learning (ICML 95)*, p. 30–37.

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

Bertsekas D. P. & Yu H. (2009). Projected equation methods for approximate solution of large linear systems. *J. Computational 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. & Barto A. G. (1996). Linear Least-Squares algorithms for temporal difference learning. *Machine Learning*, **22**(1-3), 33–57.

Choi D. & Roy B. V. (2006). A Generalized Kalman Filter for Fixed Point Approximation and Efficient Temporal-Difference Learning. *Discrete Event Dynamic Systems*, **16**, 207–239.

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

Engel Y., Mannor S. & Meir R. (2005). Reinforcement Learning with Gaussian Processes. In *Proceedings of the International Conference on Machine Learning (ICML 05)*.

Geist M. & Pietquin O. (2010a). *A Brief Survey of Parametric Value Function Approximation.* Rapport interne, Supélec.

Geist M. & Pietquin O. (2010b). Eligibility Traces through Colored Noises. In *Proceedings of the IEEE International Conference on Ultra Modern Control systems (ICUMT 2010)*, Moscow (Russia).

Geist M. & Pietquin O. (2010c). Kalman Temporal Differences. *Journal of Artificial Intelligence Research (JAIR)*, **39**, 483–532.

Kearns M. & Singh S. (2000). Bias-variance error bounds for temporal difference updates. In *In Proceedings of the 13th Annual Conference on Computational Learning Theory*, p. 142–147.

Munos R. (2003). Error bounds for approximate policy iteration. In *ICML*, p. 560–567.

Nedić A. & Bertsekas D. P. (2003). Least Squares Policy Evaluation Algorithms with Linear Function Approximation. *Discrete Event Dynamic Systems: Theory and Applications*, **13**, 79–110.

Precup D., Sutton R. S. & Singh S. P. (2000). Eligibility Traces for Off-Policy Policy Evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 00)*, p. 759–766, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

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 *27th International Conference on Machine Learning - ICML 2010*, Haïfa, Israël.

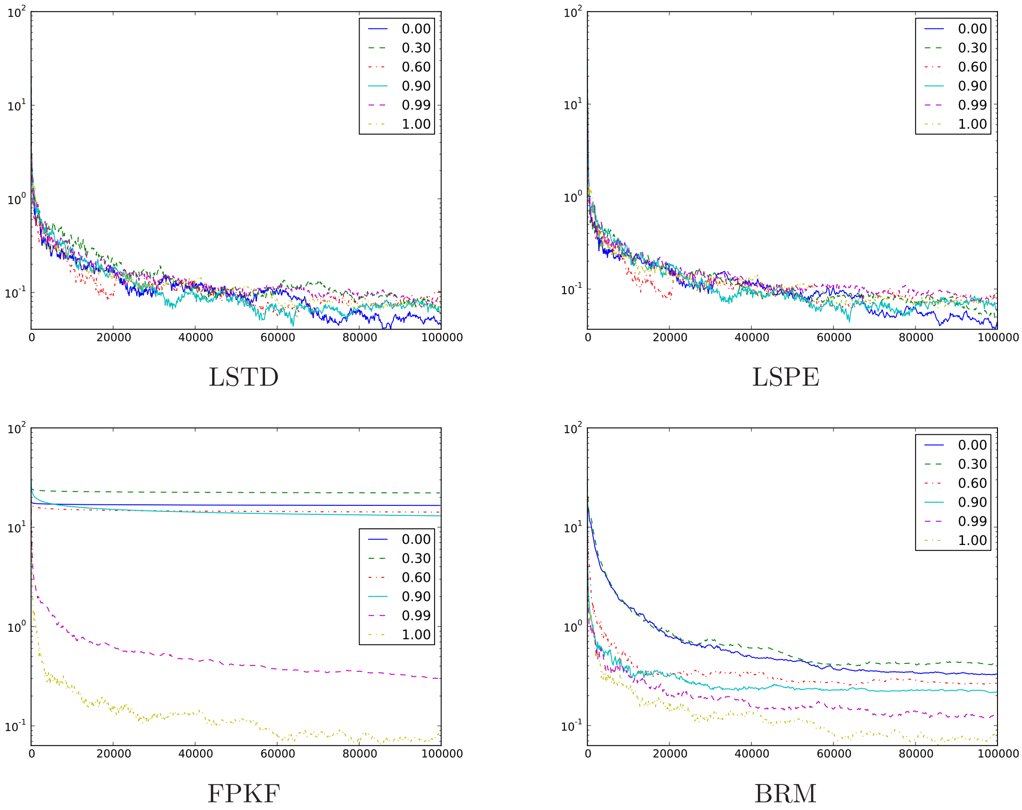
Schoknecht R. (2002). Optimality of reinforcement learning algorithms with linear function approximation. In *NIPS*, p. 1555–1562.

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

Tsitsiklis J. & 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 *Proceedings of the 27 th International Conference on Machine Learning*, Haifa, Israel.

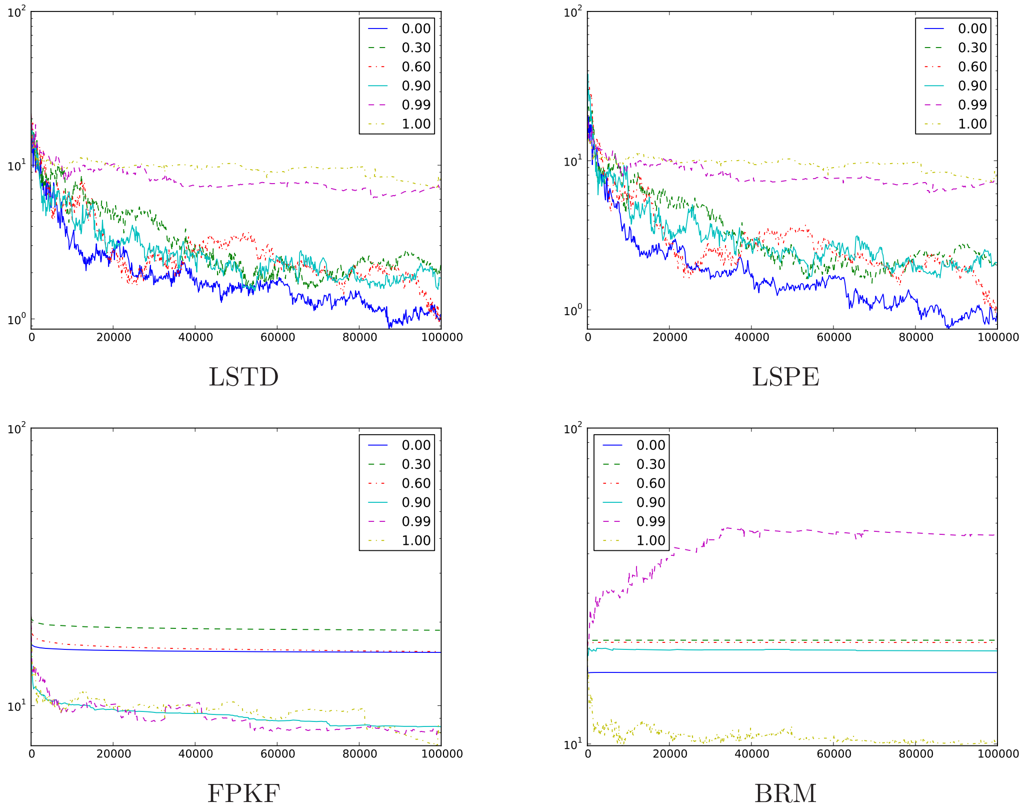
Yu H. & Bertsekas D. (2008). *New Error Bounds for Approximations from Projected Linear Equations.* Rapport interne C-2008-43, Dept. Computer Science, Univ. of Helsinki.



LSTD LSPE

FPKF BRM

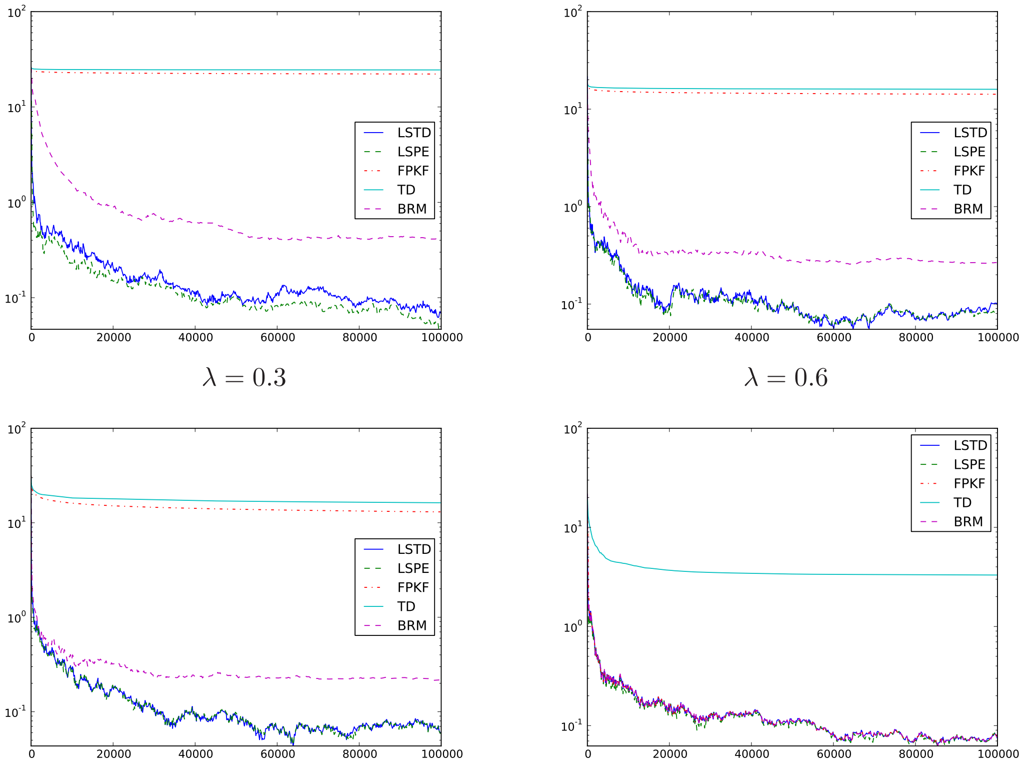
Figure 1: Influence of λ, *on-policy*



LSTD LSPE

FPKF BRM

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



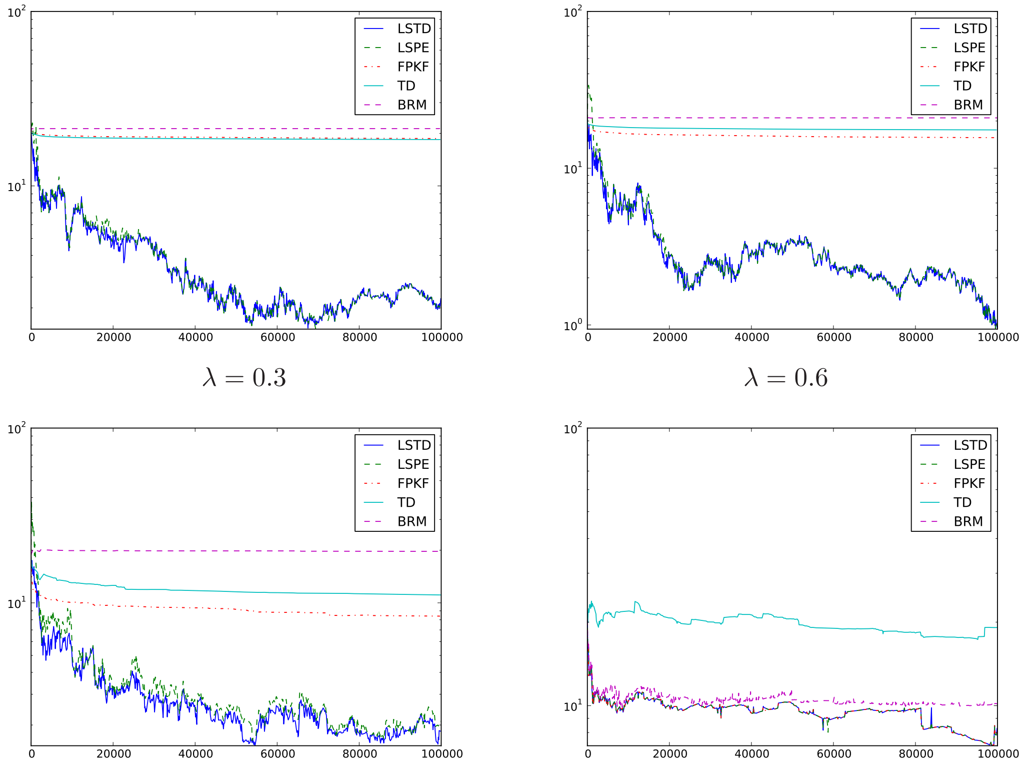
$\lambda = 0.3$

$\lambda = 0.6$

$\lambda = 0.9$

$\lambda = 1$

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



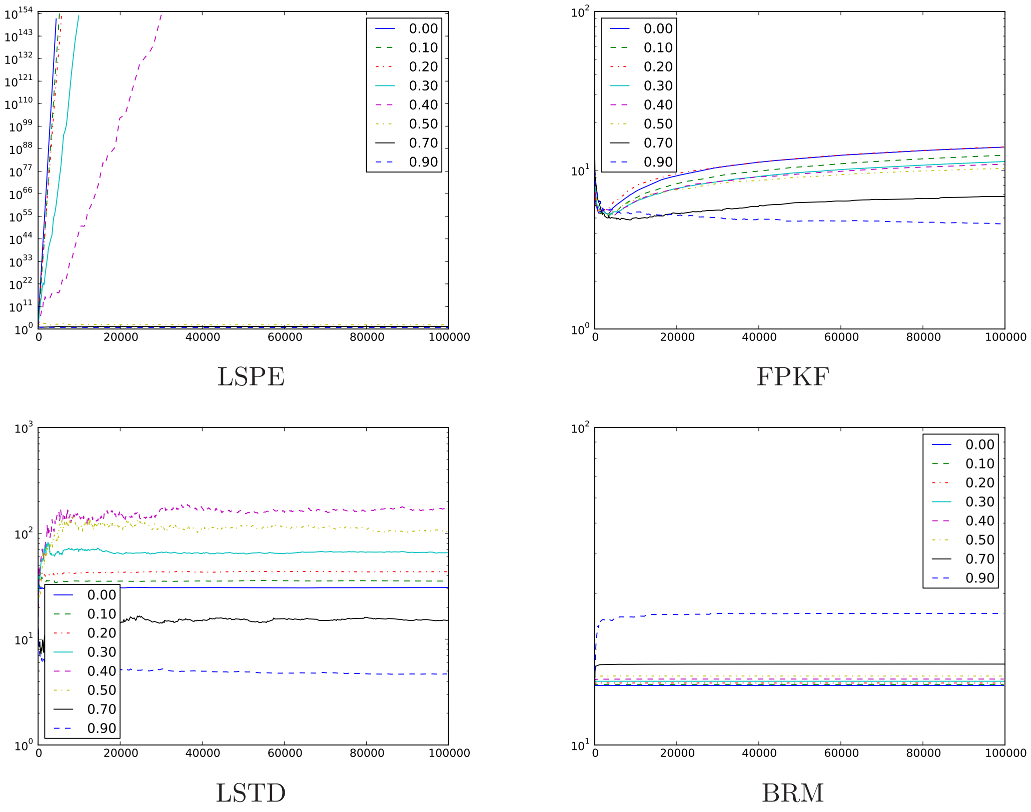
$\lambda = 0.3$

$\lambda = 0.6$

$\lambda = 0.9$

$\lambda = 1$

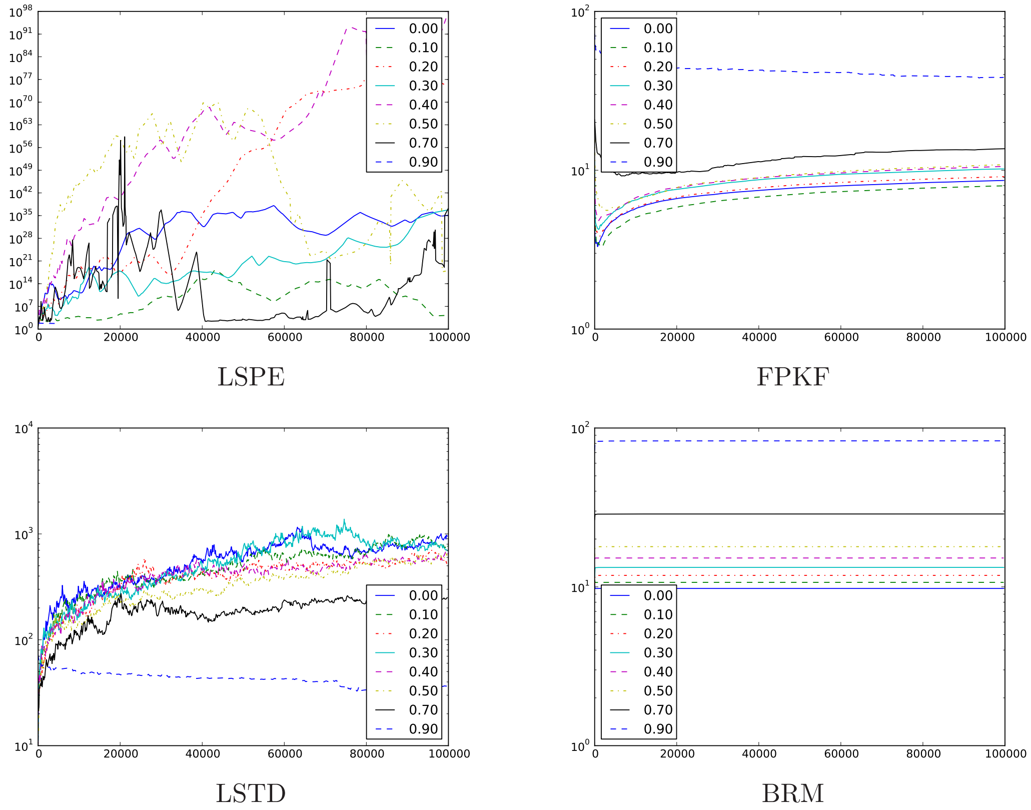
Figure 4: Comparison of the algorithms, *off-policy*



LSPE FPKF

LSTD BRM

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



LSPE FPKF

LSTD BRM

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

Appendix: Proof of Theorem 2 (Convergence of BRM(λ))

The proof of Theorem 2, which follows the general idea of that of Proposition 4 Bertsekas & Yu (2009) 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, 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 $\frac{1}{i}\tilde{b}_i \to \tilde{b}$. 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 (10) 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 $(s_{i-2l}, \dots, s_i, s_{i+1})$. Write E_0 the expectation with respect to its stationary distribution. By the Markov chain Ergodic Theorem, we have with probability 1:

$$\lim_{i\to\infty} \frac{\tilde{A}_{i,l}}{i} = E_0\left[\Delta\phi_{2l}\Delta\phi_{2l}^T y_{2l,l} + \lambda\gamma\rho_{2l-1}(\Delta\phi_{2l}\Delta_{2l-1,l}^T + \Delta_{2l-1,l}\Delta\phi_{2l}^T)\right]. \tag{11}$$

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 5 (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 $\tilde{D}_i = \operatorname{diag}\left((\tilde{P}^T)^i\mu_0\right)$. Then*

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

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,

$$\begin{aligned} E_0\left[\Delta\phi_{2l} \Delta\phi_{2l}^T y_{2l,l}\right] &= \Phi^T \left\{ \sum_{m=l+1}^{2l} (\lambda\gamma)^{2(2l-m)} \left(\tilde{D}_{2l-m} - \gamma \tilde{D}_{2l-m} P - \gamma P^T \tilde{D}_{2l-m} + \gamma^2 \tilde{D}_{2l+1-m}\right) \right\} \Phi \\ &= \Phi^T (D_l - \gamma D_l P - \gamma P^T D_l + \gamma^2 D_l') \Phi \qquad (12) \end{aligned}$$

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

Similarly, the second term on the right side of Equation (11) satisfies:

$$\begin{aligned} & E_0\left[\rho_{2l-1} \Delta_{2l-1,l} \Delta\phi_{2l}^T\right] = E_0\left[\rho_{2l-1} \sum_{k=l}^{2l-1} \tilde{\rho}_k^{2l-2} y_{k,l} \Delta\phi_k \Delta\phi_{2l}^T\right] \\ &= E_0\left[\sum_{k=l}^{2l-1} (\lambda\gamma)^{2l-1-k} \rho_k^{2l-1} \left(\sum_{m=k-l+1}^{k} (\tilde{\rho}_m^{k-1})^2\right) \Phi^T (x_k - \gamma\rho_k x_{k+1})(x_{2l} - \gamma\rho_{2l} x_{2l+1})^T \Phi \Delta\phi_{2l}^T\right] \\ &= \Phi^T \left(\sum_{k=l}^{2l-1} (\lambda\gamma)^{2l-1-k} \sum_{m=k-l+1}^{k} (\lambda\gamma)^{2(k-m)} E_0\left[\rho_m^{2l-1} \rho_m^{k-1} (x_k - \gamma\rho_k x_{k+1})(x_{2l} - \gamma\rho_{2l} x_{2l+1})^T\right]\right) \Phi \\ &= \Phi^T \left(\sum_{k=l}^{2l-1} (\lambda\gamma)^{2l-1-k} \sum_{m=k-l+1}^{k} (\lambda\gamma)^{2(k-m)} E_0\left[X_{m,k,2l} - \gamma X_{m,k+1,2l} - \gamma X_{m,k,2l+1} + \gamma^2 X_{m,k+1,2l+1}\right]\right) \Phi \\ &= \Phi^T \left(\sum_{k=l}^{2l-1} (\lambda\gamma)^{2l-1-k} \sum_{m=k-l+1}^{k} (\lambda\gamma)^{2(k-m)} \left(\tilde{D}_{k-m} P^{2l-k} - \gamma \tilde{D}_{k+1-m} P^{2l-k-1} - \gamma \tilde{D}_{k-m} P^{2l+1-k} + \gamma^2 \tilde{D}_{k+1-m} P^{2l-k}\right)\right) \Phi \\ &= \Phi^T \left(\sum_{k=l}^{2l-1} (\lambda\gamma)^{2l-1-k} \sum_{m=k-l+1}^{k} (\lambda\gamma)^{2(k-m)} \left(\tilde{D}_{k-m} P^{2l-k} (I - \gamma P) - \gamma \tilde{D}_{k+1-m} P^{2l-1-k} (I - \gamma P)\right)\right) \Phi \\ &= \Phi^T \left(\sum_{k=l}^{2l-1} (\lambda\gamma)^{2l-1-k} \sum_{m=k-l+1}^{k} (\lambda\gamma)^{2(k-m)} \left(\tilde{D}_{k-m} P - \gamma \tilde{D}_{k+1-m}\right) P^{2l-1-k} (I - \gamma P)\right) \Phi \\ &= \Phi^T \left(\sum_{k=l}^{2l-1} (\lambda\gamma)^{2l-1-k} \left(D_l P - \gamma D_l'\right) P^{2l-1-k} (I - \gamma P)\right) \Phi \\ &= \Phi^T \left(D_l P - \gamma D_l'\right) Q_l (I - \gamma P) \Phi \end{aligned}$$

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

Gathering this and Equation (12), we see that the limit of $\frac{A_{i,l}}{i}$ expressed in Equation (11) 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 (10) 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:

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

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

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