Bayesian Inference for Least Squares Temporal Difference Regularization

Nikolaos Tziortziotis, Christos Dimitrakakis

Bayesian Inference for Least Squares Temporal Difference Regularization

Nikolaos Tziortziotis[1] and Christos Dimitrakakis[2,3]

[1] LIX, École Polytechnique, F-91120 Palaiseau, France
ntziorzi@gmail.com
[2] University of Lille, F-59650 Villeneuve-dAscq, France
[3] SEAS, Harvard University, Cambridge MA-02138, USA
christos.dimitrakakis@gmail.com

Abstract. This paper proposes a fully Bayesian approach for Least-Squares Temporal Differences (LSTD), resulting in fully probabilistic inference of value functions that avoids the overfitting commonly experienced with classical LSTD when the number of features is larger than the number of samples. Sparse Bayesian learning provides an elegant solution through the introduction of a prior over value function parameters. This gives us the advantages of probabilistic predictions, a sparse model, and good generalisation capabilities, as irrelevant parameters are marginalised out. The algorithm efficiently approximates the posterior distribution through variational inference. We demonstrate the ability of the algorithm in avoiding overfitting experimentally.

1 Introduction

Value function estimation is an integral part of many reinforcement learning (RL) [29] algorithms (e.g., policy evaluation step of policy iteration) as it assesses the quality of a fixed control policy. This is straightforward in domains with a finite number of states. Large or infinite state spaces generally prohibit an explicit value function representation, but we can always represent the value function through a parameterized class of functions. In this paper we focus on the case of linear architectures where the values are approximated by a linear combination of a number of features. This approach is used by the Least-Squares Temporal Difference (LSTD) [6] algorithm, a temporal-difference algorithm that finds a linear approximation to the value function that minimizes the *mean squared projected bellman error* (MSPBE) [30].

The selection of features is critical for LSTD, as it determines the expressiveness of the value function representation. The richer the feature space is, the more likely that the value function space will contain a good approximation to the value function, but more data are needed [21]. This problem, already present in linear regression is only exacerbated in RL problems. Furthermore, using too many features makes use of the computed policies rather slow.

In linear regression, regularization is commonly used to control over-fitting, through a penalty term which discourages coefficients from reaching large values.

In regression problems, two of the most effective regularization approaches are ℓ_1 and ℓ_2-regularization [15] which involve adding a penalty term (ℓ_1 and ℓ_2 norms of the parameter vector, respectively) to the error function in order to discourage model's parameters from getting large values. In both schemes, a coefficient term λ, which typically must be selected in advance, governs the relative importance of the penalty term compared to the error function.

Bayesian reinforcement learning (BRL) (see [34] for an overview) is a framework for designing RL algorithms that models the reinforcement learning problem in a Bayesian decision theoretic manner. In *model-free* BRL, a probability distribution is maintained over the parameters of the value function, which quantifies our uncertainty over its parameters. One of the first such algorithm was Gaussian-process temporal-difference learning (GPTD) [9], which assumes that the unknown true values over the observed states are random variables generated by a Gaussian process. More specifically, GPTD incorporates a Gaussian prior over value functions and assumes a Gaussian noise model. Thus, the solution to the inference problem is given by the posterior distribution conditioned on the observed sequence of rewards. A sparse Bayesian extension of GPTD was proposed in [32,33], called RVMTD, where adopted a sparse kernelized Bayesian learning approach [31]. However, RVMTD minimizes the mean Bellman error instead of the MSPBE as in our case.

In this paper, we propose a Bayesian treatment of the LSTD algorithm, called BLSTD, that instead of seeking only a point estimate over the unknown value function parameters, actually considers the uncertainty on the value function. We adopt a fully probabilistic framework by introducing a stochastic variant of the standard Bellman operator as well as a prior distribution over the unknown model's parameters. To avoid overfitting, we further extend BLSTD algorithm with a sparse Bayesian learning approach [3,31], which we call VBLSTD. By using a tractable variational approach to automatically determine the model's complexity, we obviate the need to select a regularization parameter. We demonstrate the performance of the proposed algorithms on a number of domains, showing the ability of our model to avoid overfitting.

The remainder of the paper is organised as follows. Section 2 presents some preliminaries, review the LSTD algorithm and gives an overview of related work. Sections 3 introduce the Bayesian LSTD algorithm. In Section 4 we extend the Bayesian LSTD algorithm, presenting the VBLSTD algorithm that constitutes the main contribution of this paper. Our empirical analysis is presented in Section 5. We conclude the paper in Section 6 by discussing future directions.

2 Preliminaries and related work

A Markov Decision Process (MDP) is a tuple $\{\mathcal{S}, \mathcal{A}, P, r, \gamma\}$, where $\mathcal{S}$ is a set of states; $\mathcal{A}$ is a set of actions; $P(\cdot|s, a)$ is a transition probability kernel, defining the probability of next states in $\mathcal{S}$ for any state action pair $s \in \mathcal{S}$ and $a \in \mathcal{A}$; $r : \mathcal{S} \to \mathbb{R}$ is a reward function and $\gamma \in [0, 1]$ is a constant discount factor. The

policy $\pi : \mathcal{S} \to \mathcal{A}$ to be evaluated is a deterministic mapping from states to actions.

Value functions are of central interest in reinforcement learning. Briefly, value function V^π defines the expected discounted sum of rewards for the policy π, given that we start at state s: $V^\pi(s) \triangleq \mathbb{E}^\pi\left[\sum_{t=0}^{\infty} \gamma^t r(s_t) | s_0 = s\right]$, with $V^* \triangleq \sup_\pi V^\pi$. It is known [28] that the value function is the unique fixed-point of the Bellman operator T^π, i.e., $V^\pi = T^\pi V^\pi$, defined as:

$$(T^\pi V)(s) = r(s) + \gamma \int_{\mathcal{S}} V(s') dP(s'|s, \pi(s)), \tag{1}$$

or in a more compact form as $T^\pi V = \boldsymbol{r} + \gamma P^\pi V$, where V and $\boldsymbol{r}$ are vectors of size $|\mathcal{S}|$ that contains the state values and rewards, respectively. When the rewards and transition probabilities are known, the value function can be obtained analytically by solving the next linear system $V^\pi = (\boldsymbol{I} - \gamma P^\pi)^{-1}\boldsymbol{r}$.

In practice, however, the MDP is unknown, and we only have access to a set of n observations $\mathcal{D} = \{(s_i, r_i, s_i')\}_{i=1}^n$ generated by the policy we wish to evaluate, i.e., $s_i' \sim P(s_i, \pi(s_i))$[4]. An additional difficulty is that when the state space is large (e.g., continuous) the value function cannot be represented exactly. It is then common to use some form of parametric value function approximation. In this paper we consider linear approximation architectures with parameters $\boldsymbol{\theta} \in \mathbb{R}^k$ over k features $\boldsymbol{\phi} : \mathcal{S} \to \mathbb{R}^k$, $\boldsymbol{\phi}(\cdot) = (\phi_1(\cdot), \ldots, \phi_k(\cdot))^\top$:

$$V_{\boldsymbol{\theta}}^\pi(s) = \boldsymbol{\phi}(s)^\top \boldsymbol{\theta} = \sum_{i=1}^{k} \phi_i(s)\theta_i.$$

Throughout the paper we denote by $\mathcal{F}$ the linear function space spanned by the features ϕ_i, i.e., $\mathcal{F} = \{f_{\boldsymbol{\theta}} | f_{\boldsymbol{\theta}}(\cdot) = \boldsymbol{\phi}(\cdot)^\top \boldsymbol{\theta}\}$. Roughly speaking, $\mathcal{F}$ contains all the value functions that can be represented by the features. Let us also introduce the projection operator Π onto $\mathcal{F}$, which takes any value function $\boldsymbol{u}$ and projects it to the nearest value function, such that $\Pi\boldsymbol{u} = V_{\boldsymbol{\theta}}^\pi$ where the corresponding parameters are the solution to the least-squares problem: $\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \|V_{\boldsymbol{\theta}}^\pi - \boldsymbol{u}\|_D^2$[5] [30]. As the parameterization is linear, it is straightforward to show that the projection operator is linear and independent of the parameters $\boldsymbol{\theta}$ and given by $\Pi = \Phi C^{-1} \Phi^\top D$, where $\Phi \in \mathbb{R}^{|\mathcal{S}| \times k}$ is a matrix whose rows contain the feature vector $\boldsymbol{\phi}(s)^\top, \forall s \in \mathcal{S}$ and $C = \Phi^\top D \Phi$ is the Gram matrix.

2.1 Least Squares Temporal Difference

The least-squares temporal difference (LSTD) algorithm was introduced by Bradtke et al. [6] and computes the fixed-point of the composed projection and Bellman operators: $V_{\boldsymbol{\theta}}^\pi = \Pi T^\pi V_{\boldsymbol{\theta}}^\pi$ (see Fig. 1). It can be seen as minimizing the

[4] With the starting state $s_0 \sim d(\cdot)$ sampled from some starting distribution d.

[5] The squared norm $\|\boldsymbol{u}\|_D^2 = \boldsymbol{u}^\top D \boldsymbol{u}$ is weighted by the non-negative diagonal matrix $D \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ with elements $d(s)$ on its diagonal.

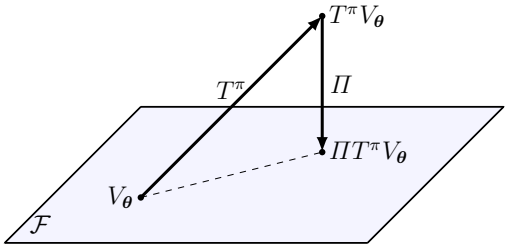


Fig. 1. A graphical representation of the LSTD problem. Here we can see the geometric relationship between the MSBE and the MSPBE. Figure adopted from [16].

mean-square projected Bellman error (MSPBE), i.e., the distance between $V_{\boldsymbol{\theta}}$ and its projected Bellman image onto $\mathcal{F}$:

$$\boldsymbol{\theta} = \underset{\boldsymbol{\theta} \in \mathbb{R}^k}{\arg\min} \, \| V_{\boldsymbol{\theta}}^\pi - \Pi T^\pi V_{\boldsymbol{\theta}}^\pi \|_D^2. \tag{2}$$

As is shown in prior work [1], LSTD seen as solving the following nested optimization problem:

$$\boldsymbol{u}^* = \underset{\boldsymbol{u} \in \mathbb{R}^k}{\arg\min} \, \| \Phi \boldsymbol{u} - T^\pi \Phi \boldsymbol{\theta} \|_D^2, \qquad \boldsymbol{\theta} = \underset{\boldsymbol{\theta} \in \mathbb{R}^k}{\arg\min} \, \| \Phi \boldsymbol{\theta} - \Phi \boldsymbol{u}^* \|_D^2, \tag{3}$$

where the first (projection) step finds the back-projection of $T^\pi V_{\boldsymbol{\theta}}^\pi$ onto $\mathcal{F}$, and the second (fixed-point) step solves the fixed-point problem which minimizes the distance between $V_{\boldsymbol{\theta}}^\pi$ and its projection.

As we discussed, usually the MDP model is unknown, or the full Φ matrices are too large to be formed, and so LSTD relies on sample-based estimates. Using a set $\mathcal{D}$ of samples from the MDP of interest, we can define $\tilde{\Phi} = [\phi(s_1)^\top; \dots; \phi(s_n)^\top]$ and $\tilde{\Phi}' = [\phi(s_1')^\top; \dots; \phi(s_n')^\top]$ to be the sampled feature matrices of successive transition states, and as $\tilde{R} = [r_i, \dots, r_n]^\top$ the sampled reward vector. Given these samples, the sample-based LSTD solution is given by the empirical version of equation (3):

$$\begin{aligned} \boldsymbol{u}^* &= \tilde{C}^{-1} \tilde{\Phi} (\tilde{R} + \gamma \tilde{\Phi}' \boldsymbol{\theta}), \\ \boldsymbol{\theta} &= (\tilde{\Phi}^\top (\tilde{\Phi} - \gamma \tilde{\Phi}'))^{-1} \tilde{\Phi}^\top \tilde{R} = A^{-1} \boldsymbol{b}, \end{aligned}$$

where we have defined

$$\tilde{C} \triangleq \tilde{\Phi}^\top \tilde{\Phi}, \quad A \triangleq \tilde{\Phi}^\top (\tilde{\Phi} - \gamma \tilde{\Phi}'), \quad \text{and} \quad \boldsymbol{b} \triangleq \tilde{\Phi}^\top \tilde{R}.$$

As the number of samples n increases, the LSTD solution $\tilde{\Phi}\boldsymbol{\theta}$ converges to the fixed-point of $\hat{\Pi} T^\pi$ [6,21,24]. We denote as $\hat{\Pi}$ the sample based feature space projector (empirical projection).

2.2 Review of Regularized LSTD Schemes

Despite the fact that LSTD offers an unbiased estimate of the value function [16], high-dimensional feature space create additional challenges. The larger the

number of features is, the more samples required to estimate $\boldsymbol{\theta}$. In some cases, the number of features may even significantly outnumber the number of observed samples $n \gg k$, leading to severe overfitting and poor prediction as the matrix A will be ill-conditioned.

For this reason, some form of regularization or model selection should be adopted, in order to prevent overfitting. Indeed, a plethora of methods have been proposed in value function approximation in RL, using different regularization and feature selection schemes (see [7] for an overview). A common form of regularization is based on ridge regression: this simply adds a term $\lambda \boldsymbol{I}$ to A, which is essentially ℓ_2-regularization. This idea was introduced and analysed by Farahmand et al. [10] for the $\ell_{2,2}$-LSTD algorithm, which uses an ℓ_2-penalty for both the projection and fixed-point steps. However, when the number of samples is much smaller than the number of features, ridge regression may fail, as it does not encourage sparsity.

On the other hand, as ℓ_1-penalties enforce sparsity, it is natural to consider those instead. The LASSO-TD variant[6] incorporates an ℓ_1-penalty in the projection step. LARS-TD [19] applies ℓ_1-regularization to the projection operator in the feature space $\mathcal{F}$, using a variant of LARS [8]. Finally, LC-TD [17] reformulates Lasso-TD as a linear complementary (LC) problem, allowing the usage of any efficient off-shelf solver. It should be emphasised that some of the solvers allow warm-starts, offering a significant computational advantage in the policy iteration context. In order for both LARS-TD and LC-TD to find a solution, matrix A is required to be a *P-matrix*[7]. The theoretical properties of the Lasso-TD problem were examined in [14], demonstrating that LARS-TD and LC-TD converge to the same solution. Particularly, it has been shown that Lasso-TD is guaranteed to have a unique fixed point. Additionally, Pires [27] suggests to solve the linear system of LSTD by including an ℓ_1-regularization term directly to it. This is a typical convex optimization problem where any standard solver can be used, being also applicable to off-policy learning.

Two closely related algorithms have been proposed in order to alleviate some of the limitations of Lasso-TD (e.g., *P-matrix* constraint), the ℓ_1-PBR (Projected Bellman residual) [11] and the $\ell_{2,1}$-LSTD [16]. Both of them place an ℓ_1-penalty term in the fix-point step, which actually penalizes the projected Bellman residual and yields a convex optimization problem. In contrast with ℓ_1-PBR, $\ell_{2,1}$-LSTD puts also an ℓ_2-penalty term on the operator problem. The Dantzig-LSTD algorithm, proposed by Geist et al. [12], integrates LSTD with the Dantzig selector, converting it into a standard linear program, that can be solved efficiently. Actually, it minimizes the sum of all parameters under the constraint that the linear system of LSTD is smaller than a predefined parameter λ in each dimension. An alternative Dantzig Selector temporal difference learning algorithm has been introduced recently by Liu et al. [22], called ODDS-TD. It is a two-stage algorithm that is also able to compute the optimal denoising matrix.

[6] Based on LASSO regression, which uses ℓ_1-regularization.

[7] *P-matrix* is a squared matrix with all of its principal minors positive (superset of the class of positive definite matrices).

3 Bayesian LSTD

In this section we present a Bayesian LSTD algorithm, called BLSTD. In our analysis, we model the fact that the transition distribution P is not known exactly by considering an *empirical Bellman operator*, given by the standard Bellman operator (1) plus additive white noise, $\epsilon \sim \mathcal{N}(0, \beta^{-1})$. For simplicity, we can assume that the noise term is state independent. Thus, the empirical Bellman operator can be written concisely as

$$\hat{T}^\pi V_{\boldsymbol{\theta}}^\pi = \boldsymbol{r} + \gamma P^\pi V_{\boldsymbol{\theta}}^\pi + N, \quad N \sim \mathcal{N}(0, \beta^{-1}\boldsymbol{I}).$$

In other words, our model says that $\boldsymbol{r} + \gamma \hat{P}^\pi V_{\boldsymbol{\theta}}^\pi$ is normally distributed with mean $\boldsymbol{r} + \gamma P^\pi V_{\boldsymbol{\theta}}^\pi$. We shall formulate a Bayesian regression model, that is based on a sample from this empirical Bellman operator.

As aforementioned, given the set of observations $\mathcal{D}$, LSTD seeks the value function parameters $\boldsymbol{\theta}$ which are invariant with respect to the composed operator $\hat{\Pi}\hat{T}^\pi$:

$$\begin{aligned} V_{\boldsymbol{\theta}}^\pi &= \hat{\Pi}\hat{T}^\pi V_{\boldsymbol{\theta}}^\pi \Leftrightarrow \\ \tilde{\Phi}^\top \tilde{R} &= \tilde{\Phi}^\top (\tilde{\Phi} - \gamma \tilde{\Phi}')\boldsymbol{\theta} + \tilde{\Phi}^\top N, \end{aligned}$$

where we have rewritten the projection operators and approximate value function in terms of the feature matrix and parameter vectors. We can now reformulate this as the following linear regression model:

$$\boldsymbol{b} = A\boldsymbol{\theta} + \tilde{\Phi}^\top N.$$

The **likelihood function** for this model is given by:

$$p(\boldsymbol{b}|\boldsymbol{\theta}, \beta) = \mathcal{N}(\boldsymbol{b}|A\boldsymbol{\theta}, \beta^{-1}\tilde{C}).$$

Taking the logarithm of the likelihood, we have

$$\ln p(\boldsymbol{b}|\boldsymbol{\theta}, \beta) = \frac{k}{2}\ln(\beta) - \frac{1}{2}\ln(|\tilde{C}|) - \frac{k}{2}\ln(2\pi) - \frac{\beta}{2}E_{\mathcal{D}}(\boldsymbol{\theta}), \tag{4}$$

where $E_{\mathcal{D}}$ corresponds to the MSPBE:

$$E_{\mathcal{D}}(\boldsymbol{\theta}) = (\boldsymbol{b} - A\boldsymbol{\theta})^\top \tilde{C}^{-1}(\boldsymbol{b} - A\boldsymbol{\theta}).$$

To complete our Bayesian model, we now introduce a **prior distribution** over the model parameters $\boldsymbol{\theta}$. Specifically, we consider a zero-mean isotropic Gaussian conjugate prior governed by a single precision parameter α,

$$p(\boldsymbol{\theta}|\alpha) = \mathcal{N}(\boldsymbol{\theta}|0, \alpha^{-1}\boldsymbol{I}).$$

Thus, we model the *parametric* uncertainty [23], which arises if the true transition probabilities and expected rewards are not known and must be estimated.

Writing only the terms from the likelihood and prior depend on the model parameters, the log of the **posterior distribution** is given by

$$\ln p(\boldsymbol{\theta}|\mathcal{D}) \propto -\frac{\beta}{2} E_{\mathcal{D}}(\boldsymbol{\theta}) - \frac{\alpha}{2}\boldsymbol{\theta}^\top \boldsymbol{\theta}. \tag{5}$$

Taking the maximization of the posterior distribution with respect to $\boldsymbol{\theta}$ is equivalent to the minimization of the MSPBE with the addition of an ℓ_2-penalty ($\lambda = \alpha/\beta$). Thus, if hyperparameter α takes a large value, the total squared length of the parameter vector $\boldsymbol{\theta}$ is encouraged to be small. Completing the squares of equation (5),

$$\begin{aligned}\ln p(\boldsymbol{\theta}|\mathcal{D}) &\propto -\frac{\beta}{2}(\boldsymbol{b} - A\boldsymbol{\theta})^\top \tilde{C}^{-1}(\boldsymbol{b} - A\boldsymbol{\theta}) - \frac{\alpha}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta} \\ &\propto -\frac{1}{2}\boldsymbol{\theta}^\top(a\boldsymbol{I} + \beta A^\top \tilde{C}^{-1} A)\boldsymbol{\theta} + \boldsymbol{\theta}^\top \beta A^\top \tilde{C}^{-1}\boldsymbol{b} + const\end{aligned}$$

we get that the posterior distribution is also Gaussian,

$$p(\boldsymbol{\theta}|\mathcal{D}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{m}, S),$$

with the covariance and mean to be given as

$$S = (\alpha\boldsymbol{I} + \beta \underbrace{A^\top \tilde{C}^{-1} A}_{\Sigma})^{-1} \text{ and } \boldsymbol{m} = \beta S A^\top \tilde{C}^{-1}\boldsymbol{b},$$

respectively, where matrix $\Sigma \triangleq A^\top \tilde{C}^{-1} A$ is always positive definite. Hence, the predictive distribution of the value function over a new state s^* is estimated by averaging the output of all possible linear models w.r.t. the posterior distribution

$$\begin{aligned}p(V_{\boldsymbol{\theta}}^{\pi}(s^*)|s^*, \mathcal{D}) &= \int_{\boldsymbol{\theta}} p(V_{\boldsymbol{\theta}}^{\pi}(s^*)|\boldsymbol{\theta}, s^*) dp(\boldsymbol{\theta}|\boldsymbol{b}, \alpha, \beta) \\ &= \mathcal{N}(V_{\boldsymbol{\theta}}^{\pi}(s^*)|\boldsymbol{\phi}(s^*)^\top \boldsymbol{m}, \boldsymbol{\phi}(s^*)^\top S \boldsymbol{\phi}(s^*)).\end{aligned}$$

An online version of our model can also be derived easily, with the posterior distribution at any phase acting as the prior distribution for the subsequent transition [2] and by using the *matrix inversion lemma* for the covariance matrix.

Maximum likelihood. For illustrative purposes, consider also a maximum likelihood approach. Restricting respect to $\boldsymbol{\theta}$, we getting the gradient of the log likelihood (4):

$$-\frac{1}{2}\nabla_{\boldsymbol{\theta}} E_{\mathcal{D}}(\boldsymbol{\theta}) = -\beta A^\top \tilde{C}^{-1}(A\boldsymbol{\theta} - \boldsymbol{b}).$$

By setting the gradient equal to zero, we get the batch LSTD solution.

In conclusion, under our model, maximum a posteriori inference corresponds to ℓ_2-regularization, while maximum likelihood inference to standard LSTD. In the next section, we propose an extension of our model that also induces sparsity.

4 Variational Bayesian LSTD (VBLSTD)

We now extend our model through a hierarchical sparse Bayesian prior, and introduce a variational approach for inference. The hope is the resulting VBLSTD algorithm will be able to avoid the over-fitting problem through regularization. For the **prior distribution** over parameter vector $\boldsymbol{\theta}$, we use an approach similar to [31] where a sparse zero-mean Gaussian prior was considered. Specifically, our prior over the model's parameters $\boldsymbol{\theta}$ is given by:

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \prod_{i=1}^{k} \mathcal{N}(\theta_i|0, \alpha_i^{-1}),$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)^\top$ are the parameters specifying our prior. Instead of selecting an arbitrary value for $\boldsymbol{\alpha}$, we select a **hyperprior** over $\boldsymbol{\alpha}$ of the form:

$$p(\boldsymbol{\alpha}) = \prod_{i=1}^{k} \mathcal{G}amma(\alpha_i|h_a, h_b),$$

where h_a, h_b are fixed parameters. The choice of the Gamma distribution for $\boldsymbol{\alpha}$ results in a marginal distribution $p(\boldsymbol{\theta})$ that is Student-t, which is known to enforce sparse representations. To complete the specification of our model, we define a Gamma hyperprior over the noise precision β:

$$p(\beta) = \mathcal{G}amma(\beta|h_c, h_d).$$

To get broad hyperpriors, we can set those parameters to some small value, e.g., $h_a = h_a = h_c = h_d = 10^{-6}$.

Bayesian inference requires the computation of the **posterior distribution** over all latent parameters $\mathcal{Z} = \{\boldsymbol{\theta}, \boldsymbol{\alpha}, \beta\}$ given the observations:

$$p(\boldsymbol{\theta}, \boldsymbol{\alpha}, \beta|\boldsymbol{b}) = \frac{p(\boldsymbol{b}|\boldsymbol{\theta}, \beta)p(\beta)p(\boldsymbol{\theta}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})}{p(\boldsymbol{b})}.$$

As the direct computation of the marginal likelihood is analytically intractable, we resort to variational inference [3,18]. This introduces a variational approximation $\mathcal{Q}(\mathcal{Z})$ to the true distribution $p(\mathcal{Z}|\boldsymbol{b})$ over the latent variables, and the problem is defined as finding the approximation closest to the true posterior distribution in terms of KL divergence. The main insight in variational methods is the following identity,

$$\ln p(\boldsymbol{b}) = \mathcal{L}(\mathcal{Q}) + \mathrm{KL}(\mathcal{Q}\|p)$$

where we have defined,

$$\mathcal{L}(\mathcal{Q}) = \int \mathcal{Q}(\mathcal{Z}) \ln \left\{ \frac{p(\mathcal{Z}, \boldsymbol{b})}{\mathcal{Q}(\mathcal{Z})} \right\}, \tag{6}$$

$$\mathrm{KL}(\mathcal{Q}\|p) = -\int \mathcal{Q}(\mathcal{Z}) \ln \left\{ \frac{p(\mathcal{Z}|\boldsymbol{b})}{\mathcal{Q}(\mathcal{Z})} \right\}. \tag{7}$$

The $\mathrm{KL}(\mathcal{Q}\|p)$ (7) represents the Kullback-Leibler divergence between the variational posterior distribution $\mathcal{Q}(\mathcal{Z})$ and the true posterior distribution $p(\mathcal{Z}|\boldsymbol{b})$ over the latent variables. As $\mathrm{KL}(\mathcal{Q}\|p) \geq 0$, it follows that $\mathcal{L}(\mathcal{Q}) \leq \ln p(\boldsymbol{b})$, which means that $\mathcal{L}(\mathcal{Q})$ is a lower bound on $\ln p(\boldsymbol{b})$. Therefore, maximizing the evidence lower bound (ELBO, see [4] for an overview) $\mathcal{L}(\mathcal{Q})$ with respect to $\mathcal{Q}$ is equivalent to minimizing the $\mathrm{KL}(\mathcal{Q}\|p)$, as the largest value of $\mathcal{L}(\mathcal{Q})$ will be achieved when the $\mathrm{KL}(\mathcal{Q}\|p)$ becomes zero.

In our problem, we consider a **variational distribution** with a factorized Gaussian form over the latent variables (c.f. *mean field theory* [26]), such that $\mathcal{Q}(\mathcal{Z}) = \mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{\theta})\mathcal{Q}_{\boldsymbol{\alpha}}(\boldsymbol{\alpha})\mathcal{Q}_{\beta}(\beta)$. Then the optimal distribution for each one of the factors can be written as:

$$\mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{m}, S) \tag{8}$$

$$\mathcal{Q}_{\beta}(\beta) = \mathcal{G}amma(\beta|\tilde{c}, \tilde{d}) \tag{9}$$

$$\mathcal{Q}_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) = \prod_{i=1}^{k} \mathcal{G}amma(\alpha_i|\tilde{a}_i, \tilde{b}_i) \tag{10}$$

where,

$$S = (diag\,\mathbb{E}[\boldsymbol{\alpha}] + \mathbb{E}[\beta]\Sigma)^{-1}, \quad \boldsymbol{m} = \mathbb{E}[\beta]SA^{\top}\tilde{C}^{-1}\boldsymbol{b},$$
$$\tilde{a}_i = h_a + \frac{1}{2}, \quad \tilde{b}_i = h_b + \frac{1}{2}\,\mathbb{E}[\theta_i^2],$$
$$\tilde{c} = h_c + \frac{k}{2}, \quad \tilde{d} = h_d + \frac{1}{2}\|\boldsymbol{b} - A\boldsymbol{m}\|^2_{\tilde{C}} + \frac{1}{2}\,\mathrm{tr}(\Sigma S).$$

The required moments can be expressed as follows:

$$\mathbb{E}[\alpha_i] = \tilde{a}_i/\tilde{b}_i, \quad \mathbb{E}[\beta] = \tilde{c}/\tilde{d}, \text{ and } \mathbb{E}[\theta_i^2] = \boldsymbol{m}_i^2 + S_{ii}.$$

The variational posterior distributions given in equations (8), (9) and (10) are then iteratively updated until convergence. As the evidence lower bound is convex with respect to each one of the factors, convergence is guaranteed.

Similarly to BLSTD, the **value function distribution** over a new state s^* can be approximated by averaging the output of all possible linear models w.r.t the variational posterior distribution $\mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{\theta})$

$$\begin{aligned} p(V^{\pi}_{\boldsymbol{\theta}}(s^*)|s^*, \mathcal{D}) &= \int_{\boldsymbol{\theta}} p(V^{\pi}_{\boldsymbol{\theta}}(s^*)|\boldsymbol{\theta}, s^*) d\mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \\ &= \mathcal{N}\left(V^{\pi}_{\boldsymbol{\theta}}(s^*) \;\middle|\; \boldsymbol{\phi}(s^*)^{\top}\boldsymbol{m}, \boldsymbol{\phi}(s^*)^{\top} S \boldsymbol{\phi}(s^*)\right). \end{aligned}$$

This gives us not only a specific mean value function, but also effectively expresses our uncertainty about what the value function is through the covariance terms.

The lower bound is interesting to look at more closely, as it is the quantity that we maximizing. Furthermore, it can be used as a convergence criterion for the variational inference. If the difference between the lower bound on two successive iterations is lower than a threshold, we assume that our model converges. Algorithm 1 provides the pseudocode of the sparse Bayesian LSTD algorithm.

Remark 1. The lower bound can be written as

$$\mathcal{L}(\mathcal{Q}) = \frac{1}{2}\ln|S| - \frac{1}{2}|\tilde{C}| + \sum_{i=1}^{k}\{\ln\Gamma(\tilde{a}_i) - \tilde{a}_i\ln\tilde{b}_i\} + \ln\Gamma(\tilde{c}) - \tilde{c}\ln\tilde{d} + \frac{k}{2}(1-\ln 2\pi) - k\ln\Gamma(h_a) + kh_a\ln h_b - \ln\Gamma(h_c) + h_c\ln h_d. \quad (11)$$

Proof. Decomposing equation (6) we obtain:

$$\mathcal{L}(\mathcal{Q}) = \mathbb{E}_{\boldsymbol{\theta},\beta}[\ln p(\boldsymbol{b}|\boldsymbol{\theta},\beta)] + \mathbb{E}_{\beta}[\ln p(\beta)] + \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\alpha}}[\ln p(\boldsymbol{\theta}|\boldsymbol{\alpha})] + \mathbb{E}_{\boldsymbol{\alpha}}[\ln p(\boldsymbol{\alpha})] - \mathbb{E}_{\boldsymbol{\theta}}[\ln\mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{\theta})] - \mathbb{E}_{\boldsymbol{\alpha}}[\ln\mathcal{Q}_{\boldsymbol{\alpha}}(\boldsymbol{\alpha})] - \mathbb{E}_{\beta}[\ln\mathcal{Q}_{\beta}(\beta)].$$

We now evaluate each term in turn.

$$\mathbb{E}_{\boldsymbol{\theta},\beta}[\ln p(\boldsymbol{b}|\boldsymbol{\theta},\beta)] = \frac{k}{2}(\psi(\tilde{c}) - \ln\tilde{d}) - \frac{k}{2}\ln 2\pi - \frac{1}{2}|\tilde{C}| - \frac{1}{2}\mathbb{E}[\beta]\{\|\boldsymbol{b} - A\boldsymbol{m}\|^2_{\tilde{C}} + \mathrm{tr}(\Sigma S)\}$$

$$\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\alpha}}[\ln p(\boldsymbol{\theta}|\boldsymbol{\alpha})] = -\frac{k}{2}\ln 2\pi - \frac{1}{2}\sum_{i=1}^{k}(\psi(\tilde{a}_i) - \ln\tilde{b}_i) - \frac{1}{2}\sum_{i=1}^{k}\mathbb{E}[\alpha_i](\boldsymbol{m}_i^2 + S_{ii})$$

$$\mathbb{E}_{\boldsymbol{\alpha}}[\ln p(\boldsymbol{\alpha})] = -k\ln\Gamma(h_a) + kh_a\ln h_b + (h_a - 1)\sum_{i=1}^{k}(\psi(\tilde{a}_i) - \ln\tilde{b}_i) - h_b\sum_{i=1}^{k}\mathbb{E}[\alpha_i]$$

$$\mathbb{E}_{\beta}[\ln p(\beta)] = -\ln\Gamma(h_c) + h_c\ln h_d + (h_c - 1)(\psi(\tilde{c}) - \ln\tilde{d}) - h_d\,\mathbb{E}[\beta]$$

$$\mathbb{E}_{\boldsymbol{\theta}}[\ln\mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{\theta})] = -\frac{1}{2}\ln|S| - \frac{k}{2}(1+\ln 2\pi)$$

$$\mathbb{E}_{\boldsymbol{\alpha}}[\ln\mathcal{Q}_{\boldsymbol{\alpha}}(\boldsymbol{\alpha})] = \sum_{i=1}^{k}\{-\ln\Gamma(\tilde{a}_i) + \tilde{a}_i\ln\tilde{b}_i + (\tilde{a}_i - 1)(\psi(\tilde{a}_i) - \ln\tilde{b}_i) - \tilde{b}_i\,\mathbb{E}[\alpha_i]\}$$

$$\mathbb{E}_{\beta}[\ln\mathcal{Q}_{\beta}(\beta)] = -\ln\Gamma(\tilde{c}) + \tilde{c}\ln\tilde{d} + (\tilde{c} - 1)(\psi(\tilde{c}) - \ln\tilde{d}) - \tilde{d}\,\mathbb{E}[\beta].$$

Substituting back, we obtain the required result. □

In the next section, we compare the Bayesian LSTD methods we derived with other state-of-the-art LSTD approaches for value function estimation.

5 Experiments

To analyze the performance of the proposed VBLSTD algorithm, we considered two discrete chain problems. Through our empirical analysis we examine both the convergence capabilities of VBLSTD on the *true* solution, as well as the ability of the VBLSTD algorithm in avoiding overfitting. In the first case, comparisons have been made with the vanilla LSTD algorithm, considering three different sizes of the Boyan's chain [5]. In the second case, comparisons have been conducted with the ℓ_2-LSTD (adding an ℓ_2-regularization factor to the projector operator), LarsTD [19] and OMPTD [25] algorithms. For that purpose, we considered the *corrupted chain problem* similar to [19,16,12].

In contrast to the VBLSTD algorithm, the performance of the three aforementioned algorithms are totally depended on the penalty parameter that should

Algorithm 1: VBLSTD$(\mathcal{D}, \boldsymbol{\phi}, \gamma)$

Initialization :
  $\max_t = 1000$, $t = 0$;
  $h_a = h_a = h_c = h_d = 10^{-6}$;
begin
  $\boldsymbol{b} = \sum_{(s,r,s')\in\mathcal{D}} \boldsymbol{\phi}(s) r$;
  $\tilde{C} = \sum_{(s,r,s')\in\mathcal{D}} \boldsymbol{\phi}(s)\boldsymbol{\phi}(s)^\top$;
  $A = \sum_{(s,r,s')\in\mathcal{D}} \boldsymbol{\phi}(s)(\boldsymbol{\phi}(s) - \gamma\boldsymbol{\phi}(s'))^\top$;
  $\Sigma = A^\top \tilde{C}^{-1} A$;
  $\langle\beta\rangle = std(\boldsymbol{b})$;
  $\langle\alpha_i\rangle = 0.01, \quad \forall i = 1, \ldots, k$;
  repeat
    $t \leftarrow t + 1$;
    1. Update $\mathcal{Q}_{\boldsymbol{\theta}}$;
      $S = (diag\, \mathbb{E}[\boldsymbol{\alpha}] + \mathbb{E}[\beta]\Sigma)^{-1}$;
      $\boldsymbol{m} = \mathbb{E}[\beta] S A^\top \tilde{C}^{-1} \boldsymbol{b}$;
      $\mathbb{E}[\theta_i^2] = \boldsymbol{m}_i^2 + S_{ii}$;
    2. Update $\mathcal{Q}_{\boldsymbol{\beta}}$;
      $\tilde{c} = h_c + k/2$;
      $\tilde{d} = h_d + \frac{1}{2}\|\boldsymbol{b} - A\boldsymbol{m}\|^2_{\tilde{C}} + \frac{1}{2}\,\mathrm{tr}(\Sigma S)$;
      $\mathbb{E}[\beta] = \tilde{c}/\tilde{d}$;
    3.Update $\mathcal{Q}_{\boldsymbol{\alpha}}$;
    for $i = 1$ **to** k **do**
      $\tilde{a}_i = h_a + 1/2$;
      $\tilde{b}_i = h_b + \mathbb{E}[\theta_i^2]/2$;
      $\mathbb{E}[\alpha_i] = \tilde{a}_i/\tilde{b}_i$;
    end
    4. Calculate bound $\mathcal{L}_t$, based on Eq.(11);
  until ***convergence or*** $t > \max_t$;
  return $\boldsymbol{m}$, S;
end

be defined explicitly in advance. Therefore, we have to answer the next question: which is the best value to set the regalurization factor? In the case of the ℓ_2-LSTD algorithm we adopted the same strategy with the one followed by Hoffman et al. [16]. Actually, we used a grid of 10 parameters logarithmically spaced between 10^{-6} and 10. In the case of the LarsTD and OMPTD algorithms, we computed the whole regularization path similar to [12] by setting the regularization factor equal to 10^{-7}. In all cases, the best prediction error has been reported.

In our experimental results, we illustrate the average root mean squared error with respect to the true value function, V^*. The optimal value function was computed explicitly since we examine discrete environments. It should be also noticed that for each run the algorithms were provided with the same rollouts of data. For each average, we also plot the 95% confidence interval for the accuracy

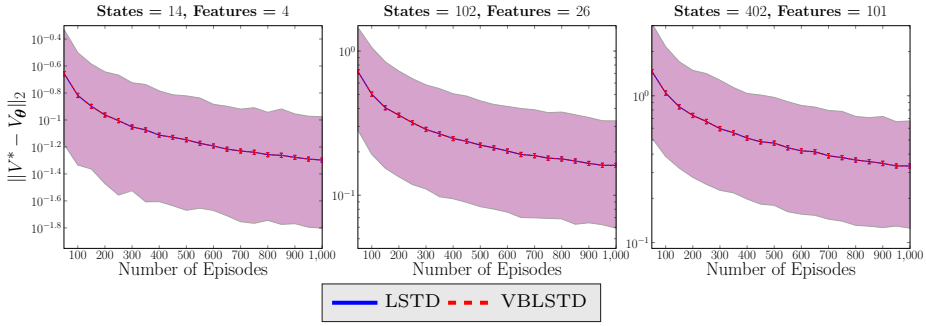


Fig. 2. Performance of policy evaluation on the Boyan's Chain for a fixed policy.

of the mean estimate with error bars. Additionally, we show the 90% percentile region of the runs, in order to indicate inter-run variability in performance.

5.1 Boyan's Chain

To demonstrate the ability of the VBLSTD algorithm to converge to the same solution with that of standard LSTD, we examine the Boyan's chain problem [5]. Actually, it is an N-state Markov chain with a single action. Each episode starts in state $N-1$ and terminates when the (absorbing) state zero is reached. For each state, $s > 2$, we transit with equal probability in states $s-1$ or $s-2$, with reward -3. On the other hand, we deterministically transit from states 2 to 1 and state 1 to 0, where the received rewards are equal to -2 and 0, respectively. Similar to [13], three different problems sizes have been considered: $N = \{14, 102, 402\}$. The feature vectors that considered for the states' representation are exactly the same with those used by Geramifard et al. [13]. Figure 2 illustrates the performance of the VBLSTD and LSTD algorithms on the three different Boyan's chain problems, averaged over 1000 runs. In all these three problems, it is clear that the proposed VBLSTD algorithm converges to same solution with the one returned by the LSTD algorithm. It means that the VBLSTD algorithm discovers the global optimum solution (i.e., the solution that corresponds to the minimum MSPBE).

5.2 Corrupted Chain

In order to examine the sparsification properties of the VBLSTD algorithm, we consider the corrupted chain problem as in [12,16,19]. This is a 20-state, 2-actions MDP proposed in [20]. In this problem, the states are connected in a chain with the actions to indicate the direction (*left* or *right*), with the probability of success to be equal to 0.9. For instance, executing *left* action at state s, the system transitions to state $s-1$ with probability 0.9 and to state $s+1$ with probability 0.1. A reward of one is given only at the ends of the chain. Similar to [12,16,19], to represent the value function we will consider $k = 6 + \overline{s}$, 6 'relevant' features (i.e., including a bias term and five RBF basis functions spaced evenly

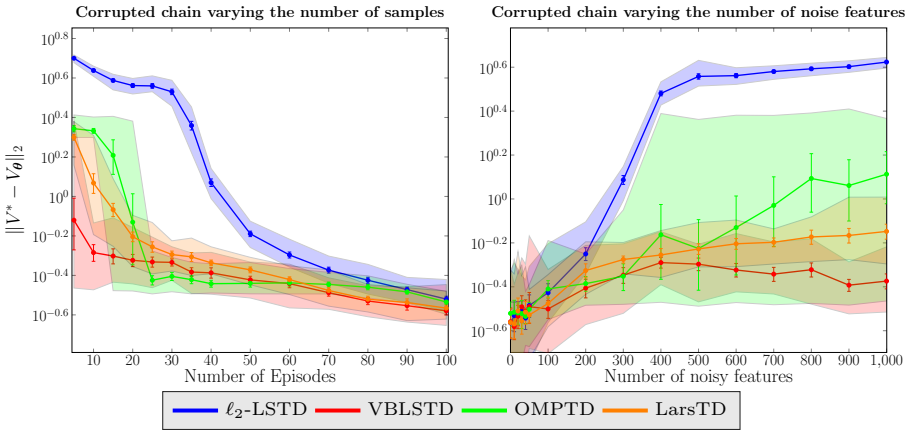
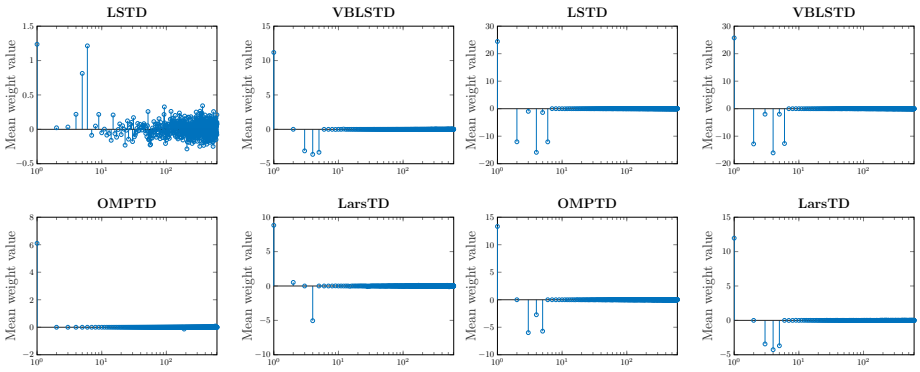


Fig. 3. Performance of policy evaluation on the Corrupted chain for a fixed policy. (Left) We consider $\overline{s} = 600$ 'irrelevant' features while varying the number of samples. The horizon of each episode is set equal to 20 steps. (Right) We use 400 transitions (20 rollouts of horizon 20) varying the number of 'irrelevant' features.

over the state space) and $\overline{s}$ additional 'irrelevant' (noise) features (containing random Gaussian noise, $\mathcal{N}(0, 1)$). It should also be stressed that through our analysis we didn't perform any standarization over the feature matrices. Also, in the case of the VBLSTD, we keep the noise precision unchanged.

The results of our experiments are presented in Fig. 3, averaged over 30 runs. We report the prediction error between the estimated and the true value function on 1000 test points. The evaluated policy is the optimal one, which selects left action on the first 10 states and right action on the rest 10. The first (left) plot shows the results in the case where we have $\overline{s} = 600$ 'irrelevant' features while varying the number of samples (the horizon of each episode is equal to 20, started randomly on $\{1, \ldots, 20\}$). On the other hand, the second (right) plot depicts the results in the case where we sample 400 transitions (20 rollouts of horizon 20), varying the number of 'irrelevant' features. In both cases, it seems that the VBLSTD algorithm performs much better compared with the others three reguralization schemes. The difference between them becomes more apparent as the number of irrelevant features increased. Additionally, it stems that the performance of VBLSTD is quite stable even when we select a large number of noise features. On the contrary, the OMPTD algorithm seems to become unstable when the number of noise features becomes large. Furthermore, we also note that the VBLSTD is not affected by overfitting when the number of features becomes greater than the number of samples. As it is expected, the performance of all algorithms is quite close even when large number of transitions are used for training or the number of noise features is quite small.

Finally, Figure 4 illustrates the mean weights (solution), $\boldsymbol{\theta}$, for each one of the examined algorithms considering 600 irrelevant features. The number of training episodes that used for training on these two plots (Fig. 4) are 10 and 100, respec-



(a) 10 Training Episodes (b) 100 Training Episodes

Fig. 4. The 606 mean weight values. The first weight is the bias term, the next 5 correspond to the relevant features (RBFs), the rest 600 correspond to the noise (irrelevant) features. The dichotomy between the 'relevant' and 'irrelevant' weights is apparent.

tively. As we can easily verify, when the number of features exceeds the number of the training samples, we encounter the overfitting problem that leads to poor predictions. It is more apparent in the case of the LSTD algorithm. However, the VBLSTD algorithm achieves to avoid the overfitting problem, succeeding to identify the relevant features even in the case where the number of samples is much lower than that of the features. Last but not least, it should be highlighted that when the number of training samples is much higher than the number of features, the solutions of the LSTD and VBLSTD algorithms are quite similar.

6 Conclusion

In this paper we introduced a fully Bayesian framework for least-squares temporal difference learning algorithm, called BLSTD. This is achieved by adopting an explicit probabilistic model for the empirical Bellman operator and introducing a prior distribution over the unknown model's parameters. This gives us the advantage of not only having a point estimate over the unknown value function parameters, but also quantifying our uncertainty about the value function. We further extended this method to a sparse variational Bayes model, called VBLSTD. The main advantage of VBLSTD compared to other regularization schemes, is its ability to avoid over-fitting by determining the models complexity in an automatic way. In practice, we verified that the VBLSTD algorithm solutions are at least as good as any other state-of-the-art algorithm, while being able to automatically ignore noisy features. We believe that this principled approach to policy evaluation can also lead to reinforcement learning algorithms with good exploration performance, something that we leave for future work.

Acknowledgements. This work was partially supported by the École Polytechnique AXA Chair (DaSciS), the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement 608743, and the Future of Life Institute.

References

1. Antos, A., Szepesvári, C., Munos, R.: Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. Machine Learning 71(1), 89–129 (2008)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
3. Bishop, C.M., Tipping, M.E.: Variational relevance vector machines. In: Uncertainty in Artificial Intelligence. pp. 46–53 (2000)
4. Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: A review for statisticians. CoRR (2016)
5. Boyan, J.: Technical update: Least-squares temporal difference learning. Machine Learning 49(2), 233–246 (2002)
6. Bradtke, S., Barto, A.: Linear least-squares algorithms for temporal difference learning. Machine Learning 22(1), 33–57 (1996)
7. Dann, C., Neumann, G., Peters, J.: Policy evaluation with temporal differences: A survey and comparison. Journal of Machine Learning Research 15, 809–883 (2014)
8. Efron, B., Hastie, T., Johnstone, L., Tibshirani, R.: Least angle regression. Annals of Statistics 32, 407–499 (2004)
9. Engel, Y., Mannor, S., Meir, R.: Reinforcement learning with gaussian process. In: International Conference on Machine Learning. pp. 201–208 (2005)
10. Farahmand, A.M., Ghavamzadeh, M., Szepesvári, C., Mannor, S.: Regularized policy iteration. In: Advances in Neural Information Processing Systems 21. pp. 441–448 (2008)
11. Geist, M., Scherrer, B.: $\ell 1$-penalized projected bellman residual. In: Recent Advances in Reinforcement Learning - 9th European Workshop. pp. 89–101 (2011)
12. Geist, M., Scherrer, B., Lazaric, A., Ghavamzadeh, M.: A dantzig selector approach to temporal difference learning. In: International Conference on Machine Learning. pp. 1399–1406 (2012)
13. Geramifard, A., Bowling, M., Sutton, R.S.: Incremental least-square temporal difference learning. In: The Twenty-first National Conference on Artificial Intelligence (AAAI). pp. 356–361 (2006)
14. Ghavamzadeh, M., Lazaric, A., Munos, R., Hoffman, M.W.: Finite-sample analysis of lasso-td. In: International Conference on Machine Learning. pp. 1177–1184 (2011)
15. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning: data mining, inference and prediction. Springer (2009)
16. Hoffman, M.W., Lazaric, A., Ghavamzadeh, M., Munos, R.: Regularized least squares temporal difference learning with nested $\ell 2$ and $\ell 1$ penalization. In: Recent Advances in Reinforcement Learning - 9th European Workshop. pp. 102–114 (2011)
17. Johns, J., Painter-wakefield, C., Parr, R.: Linear complementarity for regularized policy evaluation and improvement. In: Advances in Neural Information Processing Systems 23. pp. 1009–1017 (2010)

18. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. Machine Learning 37(2), 183–233 (1999)
19. Kolter, J.Z., Ng, A.Y.: Regularization and feature selection in least-squares temporal difference learning. In: International Conference on Machine Learning. pp. 521–528 (2009)
20. Lagoudakis, M., Parr, R.: Least-squares policy iteration. The Journal of Machine Learning Research 4, 1107–1149 (2003)
21. Lazaric, A., Ghavamzadeh, M., Munos, R.: Finite-sample analysis of LSTD. In: International Conference on Machine Learning. pp. 615–622 (2010)
22. Liu, B., Zhang, L., Liu, J.: Dantzig selector with an approximately optimal denoising matrix and its application in sparse reinforcement learning. In: Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence, UAI (2016)
23. Mannor, S., Simester, D., Sun, P., Tsitsiklis, J.N.: Bias and variance in value function estimation. In: International Conference on Machine Learning (2004)
24. Nedić, A., Bertsekas, D.P.: Least squares policy evaluation algorithms with linear function approximation. Discrete Event Dynamic Systems 13(1), 79–110 (2003)
25. Painter-Wakefield, C., Parr, R.: Greedy algorithms for sparse reinforcement learning. In: International Conference on Machine Learning (2012)
26. Parisi, G.: Statistical field theory. Frontiers in Physics, Addison-Wesley (1988)
27. Pires, B.A.: Statistical analysis of l1-penalized linear estimation with applications (2011)
28. Puterman, M.L.: Markov Decision Processes : Discrete Stochastic Dynamic Programming. John Wiley & Sons, New Jersey, US (2005)
29. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press (1998)
30. Sutton, R., Maei, H., Precup, D., Bhatnagar, S., Silver, D., Szepesvári, C., Wiewiora, E.: Fast gradient-descent methods for temporal-difference learning with linear function approximation. In: International Conference on Machine Learning. pp. 993–1000 (2009)
31. Tipping, M.E.: Sparse bayesian learning and the relevance vector machine. Journal of Machine Learning Research 1, 211–244 (2001)
32. Tziortziotis, N.: Machine Learning for Intelligent Agents. Ph.D. thesis, Department of Computer Science & Engineering, University of Ioannina, Greece (2015)
33. Tziortziotis, N., Blekas, K.: Value function approximation through sparse bayesian modeling. In: Recent Advances in Reinforcement Learning - 9th European Workshop. pp. 128–139 (2011)
34. Vlassis, N., Ghavamzadeh, M., Mannor, S., Poupart, P.: Reinforcement Learning, chap. Bayesian Reinforcement Learning, pp. 359–386. Springer (2012)