Online Bellman Residual Algorithms with Predictive Error Guarantees

Wen Sun
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
wensun@cs.cmu.edu

J. Andrew Bagnell
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
dbagnell@ri.cmu.edu

Abstract

We establish a connection between optimizing the Bellman Residual and worst case long-term predictive error. In the online learning framework, learning takes place over a sequence of trials with the goal of predicting a future discounted sum of rewards. Our analysis shows that, together with a stability assumption, *any* no-regret online learning algorithm that minimizes Bellman error ensures small prediction error. No statistical assumptions are made on the sequence of observations, which could be non-Markovian or even adversarial. Moreover, the analysis is independent of the particular form of function approximation and the particular (stable) no-regret approach taken. Our approach thus establishes a broad new family of provably sound algorithms for Bellman Residual-based learning and provides a generalization of previous worst-case result for minimizing predictive error. We investigate the potential advantages of some of this family both theoretically and empirically on benchmark problems.

1 INTRODUCTION

Reinforcement learning (RL) is an online paradigm for optimal sequential decision making where an agent interacts with an environment, takes actions, receives rewards and tries to maximize its *long-term reward*, a discounted sum of all the rewards that will be received from now on. An important part of RL is policy evaluation, the problem of evaluating the expected long-term rewards of a fixed policy. *Temporal Difference* (`TD`) is a famous family of algorithms for policy evaluation. In practice, we are typically interested in complex problem domains (e.g., continuous state space RL) and function approximations (e.g., linear functions) are used for policy evaluation. However, it has been observed that when combined with function approximation, `TD` may diverge and lead to poor prediction. The *Residual Gradient* (`RG`) was proposed (Baird, 1995) to address these concerns. `RG` attempts to minimize the *Bellman Error* (`BE`) (see definition in Sec. 2), typically with linear function approximation, using stochastic gradient descent. Since then comparison between the family of `TD` algorithms and `RG` has received tremendous attention, although most of the analyses heavily rely on certain stochastic assumptions of the environment such as that the sequence of observations are Markovian or from a static Markov Decision Process (MDP). For instance Schoknecht and Merke (2003) showed that `TD` converges provably faster than `RG` if the value functions are presented by tabular form. Scherrer (2010) shows that Bellman Residual minimization enjoys a guaranteed performance while `TD` does not in general when states are sampled from arbitrary distributions that may not correspond to trajectories taken by the system. Experimentally, they also show that `TD` converges faster but may generate poor prediction when it is close to divergence.

Schapire and Warmuth (1996) and Li (2008) provided worst-case analysis of long-term predictive error for variants of the linear `TD` and `RG` under a non-probabilistic online learning setting. Their results rely on an elegant spectral analysis of a matrix that is related to **specific** update rules of the `TD` and `RG` algorithms under linear function approximation. Unfortunately, this approach makes it more difficult to extend their worst-case (assumption free) analysis to broader families of algorithms and representations that target the Bellman and Temporal Difference errors.

Following Schapire and Warmuth (1996) and Li (2008)'s online learning framework, we present a simple, general connection between long-term predictive error and no-regret online learning that attempts to minimize `BE`. The central idea is that methods such as `RG` should be fundamentally understood as online algorithms as opposed to standard gradient methods, and that one cannot simultaneously make consistent predictions in the sense of `BE` while doing a poor job in terms of long-run predictions. Similar to Schapire and Warmuth (1996) and Li (2008), our analysis does not rely on any statistical assumptions about the underlying system. This allows us to analyze more difficult scenarios such as Markov Decision Process with transition probabilities changing over time or even with each transition chosen entirely adversarial. Our analysis generalizes to a broader class of functions to approximate the value function. Previous work from Robards et al. (2011) and Engel et al. (2005) explored the possibility of using non-linear function approximation, but to our knowledge no further analysis on the soundness with respect to prediction error are known.

Our analysis of the connection between online long-term reward prediction and no-regret online learning provides a unifying view of the relationship between prediction errors and BE and consequently suggests a broad new family of algorithms. Specifically, we present and analyze concrete examples of how to apply several well-known no-regret online algorithms such as *Online Gradient Descent* (OGD) from Zinkevich (2003), *Online Newton Step* (ONS) from Hazan et al. (2006) and *Online Frank Wolf* (OFW) from Hazan and Kale (2012) to online prediction of long-term rewards. Particularly, our analysis generalizes the RG algorithm from Baird (1995) in the following three aspects: (1) RG is a specific example of our family of algorithms that runs OGD on a sequence of BE loss functions, (2) RG can be naturally combined with more general function approximation such as functions in *Reproducing Kernel Hilbert Space* (RKHS), and (3) applying our analysis to RG provides asymptotically tighter bounds on the average prediction error of long-term rewards than that provided in Li (2008). We also find that ONS, which has no-regret rate of $O(\log T/T)$, has a faster convergence of the average prediction error of long-term rewards. With OFW, we are able to achieve sparse predictors under some conditions. We analyze these algorithms in detail in Sec. 4.

We emphasize that stability of online algorithms is essential for our results– the no-regret property can be shown by example to be insufficient to achieve low predictive error. We hence introduce the definition of *Online Stability* condition in Sec. 2, which intuitively measures the difference between two successive predictors. Our online stability condition is general enough such that most popular no-regret online algorithms naturally satisfy this condition and hence this condition does not severely limit the scope of no-regret online algorithms. Our analysis shows that the combination of the no-regret property and online stability is sufficient to promise small predictive error on the long-term rewards.

2 PRELIMINARIES

2.1 PROBLEM SETTING

We consider the sequential online learning model presented in Schapire and Warmuth (1996); Li (2008) where no statistical assumptions about the sequence of observations are made. The sequence of the observations forms a connected stream of states which can either be Markovian as typically assumed in RL problem settings or even adversarial. We define the observation at time step t as $\mathbf{x}_t \in \mathbb{R}^n$, which usually represents the features of the environment at t. Throughout the paper, we assume that feature vector $\mathbf{x}$ is bounded as $\|\mathbf{x}\|_2 \leq X, X \in \mathbb{R}^+$. The corresponding reward at step t is defined as $r_t \in \mathbb{R}$, where we assume that reward is always bounded $|r| \leq R \in \mathbb{R}^+$. Given a sequence of observations $\{\mathbf{x}_t\}$ and a sequence of rewards $\{r_t\}$, the long-term reward at t is defined as $v_t = \sum_{k=t}^{\infty} \gamma^{k-t} r_s$, where $\gamma \in [0, 1)$ is a discount factor. Given a function space $\mathcal{F}$ the learner chooses a predictor f at each time step from $\mathcal{F}$ for predicting long-term rewards. Throughout this paper, we assume that any prediction made by a predictor f at a state $\mathbf{x}$ is upper bounded as $|f(\mathbf{x})| \leq P \in \mathbb{R}^+$, for any $f \in \mathcal{F}$ and $\mathbf{x}$.

At time step $t = 0$, the learner receives $\mathbf{x}_0$, initializes a predictor $f_0 \in \mathcal{F}$ and makes prediction of v_0 as $f_0(\mathbf{x}_0)$. Rounds of learning then proceeds as follows: the learner makes a prediction of v_t at step t as $f_t(\mathbf{x}_t)$; the learner then observes a reward r_t and the next state $\mathbf{x}_{t+1}$; the learner updates its predictor to f_{t+1}. This interaction repeats and is terminated after T steps. Throughout this paper, we call this problem setting as *online prediction of long-term reward.*

We define the *signed Bellman Error* at step t for predictor f_t as $b_t = f_t(\mathbf{x}_t) - r_t - \gamma f_t(\mathbf{x}_{t+1})$, which measures effectively how self consistent f_t is in its predictions between time step t and $t+1$. For any $f^* \in \mathcal{F}$, we define the corresponding signed Bellman Error as $b_t^* = f^*(\mathbf{x}_t) - r_t - \gamma f^*(\mathbf{x}_{t+1})$. We denote the *Bellman Error* (BE) as the square of the signed Bellman error b_t^2.

The *Signed Prediction Error* of long-term reward at t for f_t is defined as $e_t = f_t(\mathbf{x}_t) - v_t$ and $e_t^* = f^*(\mathbf{x}_t) - v_t$ for f^* accordingly. We will typically be interested in bounding the *Prediction Error* (PE) e_t^2 of a given algorithm in terms of the best possible PE. To lighten notation in the following sections, all sums over time indices implicitly run from 0 to $T-1$ unless explicitly noted otherwise.

2.2 NO-REGRET ONLINE LEARNING

Under our online setting, we will define loss functional l_t at step t as the traditional Bellman Error (BE):

$$l_t(f) = (f(\mathbf{x}_t) - r_t - \gamma f(\mathbf{x}_{t+1}))^2. \qquad (1)$$

Note that $l_t(f_t) = b_t^2$.

Following the setting of online prediction of long-term reward, the learner computes predictor f_t at time step t and then receives the loss function l_t and the loss $l_t(f_t)$ (after the learner receives r_t and $\mathbf{x}_{t+1}$). We say that the online algorithm is no-regret with respect to BE if:

$$\lim_{T\to\infty} \frac{1}{T}\sum l_t(f_t) - \frac{1}{T}\sum l_t(f^*) \leq 0, \qquad (2)$$

for any predictor $f^* \in \mathcal{F}$, including the best predictor that minimizes $\sum l_t(f)$ in hindsight.

The sequence of predictors f_t being no-regret intuitively means that the predictors are giving nearly as consistent predictions over time as is possible in that function class. One might wish that the sequence of predictors being no-regret is a sufficient condition for small prediction error. More formally, one might expect that if Eq. 2 holds for the sequence of predictors $\{f_t\}$, $\sum e_t^2$ can be upper bounded:

$$\lim_{T\to\infty} \frac{1}{T}\sum e_t^2 \leq C\frac{1}{T}\sum e_t^{*2}, \quad \forall f^* \in \mathcal{F}, \qquad (3)$$

where $C \in \mathbb{R}^+$ is a constant. Schapire and Warmuth (1996) showed such a conclusion (Eq. 3) for TD and later on Li (2008) proved such a conclusion for RG, both under the assumption that $f(\mathbf{x})$ is linear.

Unfortunately, however, simply being no-regret (Eq. 2) is **not** a sufficient condition for upper bounding prediction error ($\sum e_t^2$) as in the form of Eq. 3 for general function approximation form:

Theorem 2.1 *There exists a sequence of $\{f_t\}$ that is no-regret with respect to the loss functions $\{l_t(f)\}$, but no $C \in \mathbb{R}^+$ exists that makes Eq. 3 hold.*

We prove Theorem 2.1 by providing an example in Appendix (see *Supplementary Material*) which is no-regret on $\{l_t(f_t)\}$ (Eq. 2 holds) but Eq. 3 does not hold.

2.3 ONLINE STABILITY

The counter example that supports Theorem 2.1 presents a sequence of unstable predictors $\{f_t\}$ where two successive predictors f_t and f_{t+1} vary wildly when predicting the long-term reward of $\mathbf{x}_{t+1}$. Such behavior is rather unusual for typical no-regret online learning algorithms. This suggests introducing a notion of *Online Stability* which we defined as:

Definition Online Stability: For the generated sequence of predictors f_t, we say the algorithm is online stable if:

$$\lim_{T\to\infty} \frac{1}{T}\sum (f_t(\mathbf{x}_{t+1}) - f_{t+1}(\mathbf{x}_{t+1}))^2 = 0. \qquad (4)$$

Intuitively, the online stability means that on average the difference between successive predictors is eventually small. That is, the difference between $f_t(\mathbf{x}_{t+1})$ and $f_{t+1}(\mathbf{x}_{t+1})$ is small on average. Online stability is a general condition and does not severely limit the scope of the online learning algorithms. For instance, when f is linear, the definition of stability of online learning in (Saha et al., 2012) (see Eq. 3 in Saha et al. (2012)) and (Ross and Bagnell, 2011) implies our form of online stability. We also show in the following section that many popular no-regret online learning algorithms including OGD, ONS and OWF, satisfy our online stability condition.

We show in next section that the sequence of predictors $\{f_t\}$ being no-regret with respect to the loss functions $\{l_t(f_t)\}$ **and satisfying the online stability condition** is sufficient for deriving an upper bound for prediction error as shown in Eq. 3.

3 ONLINE LEARNING FOR LONG-TERM REWARD PREDICTION

In this section, we combine the no-regret condition on loss functions $\{l_t(f)\}$ and the online stability condition together to provide a worst-case analysis of sum of PE $\sum e_t^2$, which builds a connection between the PE of long-term rewards, regret and online stability.

More formally, our worst-case analysis shows that if the online algorithm running on the sequence of loss $\{l_t(f)\}$ is no-regret and the generated sequence of predictors $\{f_t\}$ satisfies the online stability condition, predictor error can be upper bounded in the form of Eq. 3. The analysis does not place any probabilistic assumption on the sequence of observations $\{\mathbf{x}_t\}$ or any assumption on the form of predictors $f \in \mathcal{F}$ (e.g., $f(\mathbf{x})$ does not have to be linear).

We start by first providing two important lemmas below:

Lemma 3.1 *Let us define $d_t = f_t(\mathbf{x}_t) - r_t - \gamma f_{t+1}(\mathbf{x}_{t+1})$. We have:*

$$\sum d_t^2 \geq (1-\gamma)^2 \sum e_t^2 + (\gamma^2 - \gamma)(e_T^2 - e_0^2). \qquad (5)$$

Note that the difference between d_t and b_t is that d_t uses $f_{t+1}(\mathbf{x}_{t+1})$ to estimate the long-term reward at step $t+1$ while b_t uses $f_t(\mathbf{x}_{t+1})$.

Proof Schapire and Warmuth (1996) implicitly showed that $d_t = (f_t(\mathbf{x}) - v_t + v_t - (r_t + \gamma f_{t+1}(\mathbf{x}_{t+1}))) = (e_t - \gamma e_{t+1})$. Squaring both sides and summing over from $t = 0$ to $t = T-1$, we get:

$$\begin{aligned}
\sum d_t^2 &= \sum (e_t - \gamma e_{t+1})^2 \\
&= \sum e_t^2 + \gamma^2 \sum e_{t+1}^2 - 2\gamma \sum e_t e_{t+1} \\
&\geq \sum e_t^2 + \gamma^2 \sum e_{t+1}^2 - \gamma \sum e_t^2 - \gamma \sum e_{t+1}^2 \\
&= (1-\gamma)^2 \sum e_t^2 + (\gamma^2 - \gamma)(e_T^2 - e_0^2). \qquad (6)
\end{aligned}$$

The first inequality is obtained by applying Young's inequality to $2e_t e_{t+1}$ to get $2e_t e_{t+1} \leq e_t^2 + e_{t+1}^2$. ∎

Lemma 3.2 *For any $f^* \in \mathcal{F}$, the prediction error $\sum e_t^{*2}$ upper bounds the* BE $\sum b_t^{*2}$ *as follows:*

$$\sum b_t^{*2} \leq (1+\gamma)^2 \sum e_t^{*2} + (\gamma+\gamma^2)(e_0^{*2} - e_T^{*2}). \quad (7)$$

The proof of Lemma 3.2 is similar to the one for Lemma 3.1. We present the proof in Appendix.

Now let us define a measure of the change in predictors between the steps of the online algorithm as $\epsilon_t = f_t(\mathbf{x}_{t+1}) - f_{t+1}(\mathbf{x}_{t+1})$, which is closely related to the online stability condition. The b_t and d_t are then closely related with each other by ϵ_t:

$$\begin{aligned} d_t &= f_t(\mathbf{x}_t) - r_t - \gamma f_{t+1}(\mathbf{x}_{t+1}) - \gamma f_t(\mathbf{x}_{t+1}) + \gamma f_t(\mathbf{x}_{t+1}) \\ &= b_t + \gamma\epsilon_t. \end{aligned}$$

Squaring both sides, we get:

$$\begin{aligned} d_t^2 &= b_t^2 + 2b_t\gamma\epsilon_t + \gamma^2\epsilon^2 \leq b_t^2 + b_t^2 + \gamma^2\epsilon_t^2 + \gamma^2\epsilon_t^2 \\ &= 2b_t^2 + 2\gamma^2\epsilon_t^2, \end{aligned} \quad (8)$$

where the first inequality is coming from applying Young's inequality to $2b_t\gamma\epsilon_t$ to get $2b_t\gamma\epsilon_t \leq b_t^2 + \gamma^2\epsilon_t^2$. We are now ready to state the following main theorem of this paper:

Theorem 3.3 *Assume a sequence of predictors $\{f_t\}$ is generated by running some online algorithm on the sequence of loss functions $\{l_t\}$. For any predictor $f^* \in \mathcal{F}$, the sum of prediction errors $\sum e_t^2$ can be upper bounded as:*

$$\begin{aligned} (1-\gamma)^2 \sum e_t^2 \leq& 2\sum(b_t^2 - b_t^{*2}) + 2\gamma^2 \sum \epsilon_t^2 \\ &+ 2(1+\gamma)^2 \sum e_t^{*2} + M, \end{aligned} \quad (9)$$

where

$$M = 2(\gamma+\gamma^2)(e_0^{*2} - e_T^{*2}) - (\gamma^2 - \gamma)(e_T^2 - e_0^2).$$

By running a no-regret and online stable algorithm on the loss functions $\{l_t(f)\}$, as $T \to \infty$, the average prediction error is then asymptotically upper bounded by a constant factor of the best possible prediction error in the function class:

$$\lim_{T\to\infty} : \frac{\sum e_t^2}{T} \leq \frac{2(1+\gamma)^2}{(1-\gamma)^2} \frac{\sum e_t^{*2}}{T}. \quad (10)$$

Proof Combining Lemma. 3.1 and Lemma. 3.2, we have:

$$\begin{aligned} &\sum d_t^2 - 2\sum b_t^{*2} \\ &\geq (1-\gamma)^2 \sum e_t^2 + (\gamma^2 - \gamma)(e_T^2 - e_0^2) \\ &- 2(1+\gamma)^2 \sum e_t^{*2} \\ &- 2(\gamma+\gamma^2)(e_0^{*2} - e_T^{*2}). \end{aligned} \quad (11)$$

Subtracting $2b_t^{*2}$ on both sides of Eq. 8, and then summing over from $t = 1$ to $T-1$, we have:

$$\sum d_t^2 - \sum 2b_t^{*2} \leq 2\sum(b_t^2 - b_t^{*2}) + 2\gamma^2 \sum \epsilon_t^2.$$

Combining the above two inequalities together, we have:

$$\begin{aligned} &2\sum(b_t^2 - b_t^{*2}) + 2\gamma^2 \sum \epsilon_t^2 \\ &\geq (1-\gamma)^2 \sum e_t^2 + (\gamma^2 - \gamma)(e_T^2 - e_0^2) \\ &- 2(1+\gamma)^2 \sum e_t^{*2} - 2(\gamma+\gamma^2)(e_0^{*2} - e_T^{*2}). \end{aligned} \quad (12)$$

Rearrange inequality (12) and define $M = 2(\gamma+\gamma^2)(e_0^{*2} - e_T^{*2}) - (\gamma^2 - \gamma)(e_T^2 - e_0^2)$, we obtain inequality (9).

Assume that the $\bar{f} = \arg\min_{f\in\mathcal{F}} \sum l_t(f)$, then if the online algorithm is no-regret, we have

$$\begin{aligned} \frac{1}{T}\sum b_t^2 - b_t^{*2} &= \frac{1}{T}\sum l_t(f_t) - l_t(f^*) \\ &\leq \frac{1}{T}\sum l_t(f_t) - l_t(\bar{f}) \\ &= \frac{1}{T}\text{Regret} \leq 0, \ \ T \to \infty. \end{aligned} \quad (13)$$

If the online algorithm satisfies the stability condition (Eq. 4), we have: $\frac{1}{T}\sum \epsilon_t^2 = 0$ when $T \to \infty$.

Also, since we assume $|f(\mathbf{x})| \leq P$ and $|r| \leq R$, we can see M must be upper bounded by some constant. Hence, we must have $\frac{M}{T} = 0$, as $T \to \infty$.

Under the conditions that the online algorithm is no-regret and satisfies online stability, we get Eq. 10 by dividing both sides of Eq. 9 by T and taking T to infinity. ∎

Note that in Theorem 3.3, Eq. 9 holds for any $f^* \in \mathcal{F}$, including the f^* that minimizes the prediction error. But note that the one that minimizes prediction error, PE, does not necessarily optimize the BE, which may lead to an improvement of the bound in Eq. 10 in practice. To see this, note that we showed in the proof that for a no-regret algorithm:

$$\frac{1}{T}\sum b_t^2 - b_t^{*2} \leq \frac{1}{T}\text{Regret} \leq 0, \text{ as } T \to \infty. \quad (14)$$

Hence the limit of $(1/T)\sum(b_t^2 - b_t^{*2})$ may be negative for some $f^* \in \mathcal{F}$, which could lead to a potential decrease in the upper bound of $(1/T)\sum e_t^2$ in Eq. 10 and give us a tighter bound in practice.

When $e_t^* = 0, \forall t$, from Theorem 3.3, it is easy to see that no-regret rate of $(1/T)\sum(b_t^2 - b_t^{*2})$ and the online stability rate of $(1/T)\sum \epsilon_t^2$ together determine the rate of the convergence of $(1/T)\sum e_t^2$.

When $T \to \infty$ and $\gamma \to 1$ (specifically when $\gamma \geq (1/\sqrt{2})$), our upper bound analysis in Eq. 10 is asymptotically tighter than the upper bound in Li (2008) (Eq. 12) provided for RG, which is a special case of our approach as we demonstrate in the following section. As we will additionally show, a large number of popular no-regret online algorithms also satisfy the online stability condition, broadening the family of algorithms that can be used to learn predictors of long-term rewards.

4 ALGORITHMS

Our analysis in Sec. 3 provides a reduction from the online prediction of long-term reward to the no-regret online learning setting on a sequence of loss functions $\{l_t(f)\}$ defined in Sec. 2.1, which enables us to develop a new set of algorithms. In this section, we give concrete examples of new Bellman Residual algorithms based on well-known no-regret online learning procedures such as Online Gradient Descent (OGD), Online Newton Step (ONS) and the Online variant of Frank Wolfe (OFW). The choice of algorithm depends on the size and sparsity level of features and the available computational budget. For instance, OGD generalizes RG and has $O(n)$ computational complexity at every update step which makes it suitable for applications where sampling observations is cheap (e.g., RL for video games). ONS provides a logarithmic no-regret rate and could lead to faster convergence in practice, making it potentially suitable for applications where obtaining samples of observations is expensive (e.g., RL for a physical robot). Finally, OFW introduces sparsity and can be applied to problems where the feature dimension is larger than the number of samples.

Although the analysis in Sec. 3 does not place any assumption on predictors $f \in \mathcal{F}$, in practice to achieve the no-regret property on the loss functions $\{l_t(f)\}$, additional assumptions of loss functions (e.g., convexity) are needed. Since we discuss concrete no-regret online algorithms in this section, for $\mathcal{F}$ we focus on vector spaces equipped with inner product. Specifically, we focus on two vector spaces: (1) *Reproducing Kernel Hilbert Spaces* (RKHS) where $f = \sum \alpha_i K(\mathbf{x}_i, \cdot) \in \mathcal{F}$, for some kernel $K(\mathbf{x}, \cdot)$ and (2) spaces consisting of linear functions $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$, $\mathbf{w} \in \mathcal{W}$[1]. We now summarize the assumptions that we will use in section:

1. We assume $\mathcal{F}$ (or $\mathcal{W}$) is convex and bounded in a sense that the diameter of $\mathcal{F}$ (or $\mathcal{W}$) is upper bounded as $\max_{f_1,f_2\in\mathcal{F}} \|f_1 - f_2\| \leq D \in \mathbb{R}^+$, where the norm $\|f\|$ is defined by the inner product associated with the function space: $\|f\|^2 = \langle f, f\rangle$;
2. We assume that $\|\mathbf{x}_t\|_2 \leq X$, $\forall t$, $|K(\mathbf{x}_1, \mathbf{x}_2)| \leq K, \forall \mathbf{x}_1, \mathbf{x}_2$, $\|f\| \leq F$, $\forall f \in \mathcal{F}$, and $\|\mathbf{w}\|_2 \leq W$, $\forall \mathbf{w} \in \mathcal{W}$, where $K \in \mathbb{R}^+, F \in \mathbb{R}^+, W \in \mathbb{R}^+$.

Any prediction $f(\mathbf{x})$ is always bounded, since for RKHS, $f(\mathbf{x})$ is bounded as $|f(\mathbf{x})| \leq \|f\|\|K(\mathbf{x}, \cdot)\| \leq F\sqrt{K}$, and for $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$, we also have $|f(\mathbf{x})| \leq \|\mathbf{w}\|_2\|\mathbf{x}\|_2 \leq WX$. For notation simplicity, we then simply assume that $f(\mathbf{x})$ is always bounded as $|f(\mathbf{x})| \leq P, P \in \mathbb{R}^+$.

Lemma 4.1 *With the above assumptions, for any pair of* $\mathbf{x}_t$ *and* $\mathbf{x}_{t+1}$*, for RKHS, the loss functional* $l_t(f)$ *is convex and Lipschitz continuous with respect to the norm defined by the inner product* $\langle\cdot,\cdot\rangle_K$*; for* $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$*, the loss function* $l_t(\mathbf{w})$ *is convex and Lipischitz continuous with respect to either* L_1 *norm* $\|\cdot\|_1$ *or* L_2 *norm* $\|\cdot\|_2$.

We present the proof of the above lemma in Appendix.

4.1 GRADIENT-BASED APPROACHES

Before diving into the detailed examples of mirror descent and gradient based approaches, we first introduce an important lemma about the stability of one particular online algorithm: *Follow the Regularized Leader* (FTRL). It is well known that gradient-based and mirror descent approaches can be understood in the framework of FTRL. In particular, we only focus on the case where the loss functions are convex and L-Lipschitz continuous with a regularization that is strongly-convex. We refer reader to Shalev-Shwartz (2011) for detailed definitions of convex functions, Lipschitz continuous, and strong convexity.

The update rule of FTRL at step t can be summarized as:

$$f_{t+1} = \arg\min_{f\in\mathcal{F}} \sum_{i=0}^{t} l_i(f) + \frac{1}{\mu}R(f). \tag{15}$$

Lemma 4.2 *For FRTL with convex and L-Lipschitz continuous loss functions* $l_t(f)$ *and strongly convex regularization function* $R(f)$ *(with respect to* $\|f\|$*), we have:*

$$\sum \|f_t - f_{t+1}\| \leq LT\mu. \tag{16}$$

Setting $\mu = \frac{1}{\sqrt{T}}$ *to achieve no-regret property, then we have:*

$$\lim_{T\to\infty} \frac{1}{T}\sum \|f_t - f_{t+1}\| = 0. \tag{17}$$

Similar proofs has been shown in (Ross and Bagnell, 2011) and (Saha et al., 2012). For completeness, we present the proof of the above lemma in Appendix following our notation and problem setting.

4.1.1 Gradient Descent on BE

Gradient descent approaches can be understood in the FTRL framework where the convex loss functions in FTRL are replaced by a linear approximation:

$$f_{t+1} = \arg\min_{f\in\mathcal{F}} \sum_{i=0}^{t} \langle g_i, f\rangle + \frac{1}{\mu_t}R(f), \tag{18}$$

[1] Linear function space is a special case of RKHS. We discuss linear function separately since some online algorithms discussed here only work for linear functions

where $g_t \in \partial l_t(f_t)$ is a sub-gradient of l_t at f_t and $R(f)$ is a strongly convex regularizer.

We first consider the special case where $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ is linear. Note that our loss function is $l_t(\mathbf{w}) = (\mathbf{w}^T\mathbf{x}_t - r_t - \gamma\mathbf{w}^T\mathbf{x}_{t+1})^2$ and its gradient g_t at $\mathbf{w}_t$ is $g_t = (\mathbf{w}_t^T\mathbf{x}_t - r_t - \gamma\mathbf{w}_t^T\mathbf{x}_{t+1})(\mathbf{x}_t - \gamma\mathbf{x}_{t+1})$. Setting the regularization $R(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_2^2$, which is 1-strongly convex, we obtain the RG algorithm in Baird (1995), where the update step at t is:

$$\mathbf{w}_{t+1} := \mathbf{w}_t - \mu_t(\mathbf{w}_t^T(\mathbf{x}_t - \gamma\mathbf{x}_{t+1}) - r_t)(\mathbf{x}_t - \gamma\mathbf{x}_{t+1}),$$

Note that the linear loss function is convex and Lipschitz continuous ($\|g_t\|_2$ is bounded based on our assumptions that $\|\mathbf{x}\|_2$, $|r|$ and $\|\mathbf{w}\|_2$ are all bounded). Online Gradient Descent (OGD) is no-regret (Zinkevich, 2003) with $\mu_t = 1/\sqrt{T}$ and from Lemma 4.2, we have:

$$\begin{aligned}&\frac{1}{T}\sum(\mathbf{w}_t^T\mathbf{x}_{t+1} - \mathbf{w}_{t+1}^T\mathbf{x}_{t+1})^2\\ &\le \frac{1}{T}\sum\|\mathbf{x}_{t+1}\|_2^2\|\mathbf{w}_t - \mathbf{w}_{t+1}\|_2^2\\ &\le X^2\frac{1}{T}\sum\|\mathbf{w}_t - \mathbf{w}_{t+1}\|_2^2 = 0,\ \ T\to\infty, \qquad (19)\end{aligned}$$

which exactly satisfies the online stability condition. Hence, RG enjoys the guarantee of our main theorem 3.3.

4.1.2 Exponentiated Gradient Descent on BE

When we set the regularization $R(\mathbf{w}) = \sum_i \mathbf{w}^i(\log(\mathbf{w}^i) - 1)$, where for vector $\mathbf{w}$, $\mathbf{w}^i$ is the i-th component of $\mathbf{w}$, we generalize RG to *Exponentiated Gradient* (EG) descent as Precup and Sutton (1997) did for TD.

Since we assumed that $\|\mathbf{w}\|_2 \le W$, then $\|\mathbf{w}\|_1 \le W'$ for $W' \in \mathbb{R}^+$. Then the regularization $R(\mathbf{w})$ becomes $(1/W')$-strongly-convex and the loss function $l_t(\mathbf{w})$ is Lipschitz continuous with respect to L_1 norm $\|\cdot\|_1$. Solving Eq.18, we obtain the update step at t as:

$$\mathbf{w}_{t+1}^i = \mathbf{w}_t^i\exp\big(-\mu_t(\mathbf{w}_t^T(\mathbf{x}_t - \gamma\mathbf{x}_{t+1}) - r_t)(\mathbf{x}_t^i - \gamma\mathbf{x}_{t+1}^i)\big).$$

Similar to RG, using Lemma 4.2 (the norm in Lemma 4.2 becomes L_1 norm) we can show that EG satisfies our online stability condition:

$$\begin{aligned}&\frac{1}{T}\sum(\mathbf{w}_t^T\mathbf{x}_{t+1} - \mathbf{w}_{t+1}^T\mathbf{x}_{t+1})^2\\ &\le \frac{1}{T}X^2\sum\|\mathbf{w}_t - \mathbf{w}_{t+1}\|_2^2\\ &\le \frac{1}{T}X^2\sum\|\mathbf{w}_t - \mathbf{w}_{t+1}\|_{\mathbf{1}}^2 = 0,\ \ T\to\infty, \qquad (20)\end{aligned}$$

and is also no-regret on $\{l_t(\mathbf{w})\}$ when $\mu_t = 1/\sqrt{T}$. Hence, EG descent approach enjoys our main theorem 3.3.

4.1.3 Gradient Descent in RKHS

Now we consider functions $f(\mathbf{x})$ that belongs to RKHS $\mathcal{H}_{\mathcal{K}}$. For the special case where $R(f) = \frac{1}{2}\langle f, f\rangle_K$, we obtain an RG-style update based on functional gradient descent (Scholkopf and Smola, 2001):

$$\begin{aligned}f_{t+1} := f_t - \mu_t\Big(&(f_t(\mathbf{x}_t) - r_t - \gamma f_t(\mathbf{x}_{t+1}))\\ &\times(K(\mathbf{x}_t,\cdot) - \gamma K(\mathbf{x}_{t+1},\cdot))\Big). \qquad (21)\end{aligned}$$

Similarly, it is straightforward to show that gradient descent in RKHS satisfies our stability condition and no-regret condition when $\mu_t = 1/\sqrt{T}$. Hence, gradient descent in RKHS also enjoys the predictive error guarantees derived in Theorem 3.3.

4.2 IMPLICIT ONLINE LEARNING

Recently Tamar et al. (2014) has demonstrated implicit online learning for temporal difference method. We consider the same approach for Bellman Residual minimization by applying implicit online learning from Kulis et al. (2010) to the loss functions $\{l_t(f)\}$ and thus provide worst-case guarantees. Specifically at iteration t, f_{t+1} is computed implicitly as:

$$\begin{aligned}f_{t+1} &= \arg\min_{f\in\mathcal{F}}\mathcal{D}_R(f, f_t) + \mu_t l_t(f) \qquad (22)\\ &= \arg\min_{f\in\mathcal{F}}\mathcal{D}_R(f, f_t) + \mu_t(f(x_t) - \gamma f(x_{t+1}) - r_t)^2,\end{aligned}$$

where $\mathcal{D}_R$ is a Bregman divergence. Saha et al. (2012) show that when $\mu_t = 1/\sqrt{t}$, the generating function $R(f)$ is positive and strongly-convex, and the loss function $l_t(f)$ is convex and Lipschitz continuous, implicit online learning is shown to be no-regret and also satisfies Eq. 17.

4.2.1 Implicit Online Gradient Descent

Particularly, we first consider $f = \sum\alpha_i K(\mathbf{x}_i,\cdot)$ in RKHS. Setting $R(f) = \frac{1}{2}\langle f, f\rangle_k$, we have $\mathcal{D}_R(f, f_t) = \frac{1}{2}\|f - f_t\|^2$. Then solving Eq. 22, we obtain the following update rule:

$$\begin{aligned}f_{t+1} := f_t - &\frac{\mu_t}{1 + \mu_t\|K(\mathbf{x}_t,\cdot) - \gamma K(\mathbf{x}_{t+1},\cdot)\|^2}\\ &\times(f_t(\mathbf{x}_t) - \gamma f_t(\mathbf{x}_{t+1}) - r_t)\\ &\times(K(\mathbf{x}_t,\cdot) - \gamma K(\mathbf{x}_{t+1},\cdot)).\end{aligned}$$

When considering linear function $f(x) = \mathbf{w}^T\mathbf{x}$ and $R(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_2^2$, we obtain a similar update step:

$$\begin{aligned}\mathbf{w}_{t+1} = \mathbf{w}_t - &\frac{\mu_t}{1 + \mu_t\|\mathbf{x}_t - \gamma\mathbf{x}_{t+1}\|^2}\\ &\times(\mathbf{w}_t^T(\mathbf{x}_t - \gamma\mathbf{x}_{t+1}) - r_t)(\mathbf{x}_t - \gamma\mathbf{x}_{t+1}).\end{aligned}$$

As we will show in the experiments, compared to RG (OGD on BE), implicit OGD on BE is less sensitive to the choice of step-size, which enables us to set large step-size to achieve faster convergence for $(1/T)\sum e_t^2$. This phenomenon is also observed by Tamar et al. (2014) when they compare implicit temporal difference to the original `TD` algorithm (Sutton and Barto, 1998).

4.3 ONLINE NEWTON STEP

We also analyze an online second-order method: the Online Newton Step (ONS) (Hazan et al., 2006) for online prediction of long-term reward. For ONS, we only focus on linear function approximation $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. We slightly adapt the ONS for our loss function $l_t(\mathbf{w})$. We first present the following lemma:

Lemma 4.3 *For loss function $l_t(\mathbf{w}) = (\mathbf{w}^T\mathbf{x}_t - r_t - \gamma\mathbf{w}^T\mathbf{x}_{t+1})^2$, there exists a $\lambda \in \mathbb{R}^+$, such that for all $\mathbf{w}$ and $\mathbf{w}'$:*

$$\begin{aligned} l_t(\mathbf{w}) \geq & l_t(\mathbf{w}') + \nabla l_t(\mathbf{w}')^T(\mathbf{w}-\mathbf{w}') \\ & + \frac{\lambda}{2}(\mathbf{w}-\mathbf{w}')^T\nabla l_t(\mathbf{w}')\nabla l_t(\mathbf{w}')^T(\mathbf{w}-\mathbf{w}'). \end{aligned}$$

We present the proof of the above lemma in appendix.

With λ, then the iterative update rule for ONS is:

$$\mathbf{w}_t = \Pi_{\mathcal{W}}^{A_{t-1}}\left(\mathbf{w}_{t-1} - \frac{1}{\lambda}A_{t-1}^{-1}\nabla l_{t-1}(\mathbf{w}_{t-1})\right), \quad (23)$$

where $A_t = \sum_{i=0}^t \nabla l_t(\mathbf{w}_t)\nabla l_t(\mathbf{w}_t)^T + \epsilon I_n$, $\epsilon \in R^+$, and $\Pi_{\mathcal{W}}^{A_t}$ is a projection to $\mathcal{W}$ with the norm induced by A_t: $\Pi_{\mathcal{W}}^{A_t}(\mathbf{y}) = \arg\min_{\mathbf{w}\in\mathcal{W}}(\mathbf{w}-\mathbf{y})^T A_t(\mathbf{w}-\mathbf{y})$, which makes this projection operator $\Pi_{\mathcal{W}}^{A_t}$ not trivial and equal to solving a convex program usually.

Since $\|\mathbf{x}\|_2 \leq X$, $\|\mathbf{w}\|_2 \leq W$ and $|r| \leq R$, we have $\|\nabla l_t(\mathbf{w})\|_2 \leq G$, for $G \in \mathbb{R}^+$. The following lemma shows that ONS satisfies the our online stability condition:

Lemma 4.4 *The sequence $\{\mathbf{w}_t\}$ generated by ONS satisfies the online stability condition:*

$$\begin{aligned} & \frac{1}{T}\sum(\mathbf{w}_t^T\mathbf{x}_{t+1} - \mathbf{w}_{t+1}^T\mathbf{x}_{t+1})^2 \\ & \leq \frac{1}{T}\frac{X^2}{G^2\lambda^2}n\log(T+1) = 0, \;\; T\rightarrow\infty. \quad (24) \end{aligned}$$

The proof borrows ideas from Hazan et al. (2006) and is presented in the Appendix. Note that the convergence rate of $\frac{1}{T}\sum(f_t(\mathbf{x}_{t+1}) - f_{t+1}(\mathbf{x}_{t+1}))^2$ is $O(\log T/T)$, which is the same as the no-regret rate of ONS.

4.4 PROJECTION-FREE ONLINE LEARNING

We analyze the Online Frank Wolfe (OFW) (Hazan and Kale, 2012) for online prediction of long-term reward. Previously introduced methods, including OGD, EG, OGD in RKHS, Implicit OGD, and implicit OGD in RKHS, usually need a projection operation in each update step if the newly updated predictor is out of its pre-defined convex set $\mathcal{F}$, although in many cases, projection operations are simple[2]. OFW is a projection-free online method and every step involves solving a (typicaly very simple) linear programming problem. Again, we restrict our analysis to linear functions $f = \mathbf{w}^T\mathbf{x}$ for OFW. Without loss of generality, we assume $l_t(\mathbf{w})$ is L-Lipschitz continuous with some $L \in \mathbb{R}^+$ (Lemma 4.1).

Applying the adversarial variant of OFW to the sequence of loss functions $\{l_t(f)\}$, we have the following iterative update step:

$$\begin{aligned} & \mathbf{v}_t = \arg\min_{\mathbf{w}\in\mathcal{W}} \nabla F_t(\mathbf{w}_t)^T\mathbf{w}, \\ & \mathbf{w}_{t+1} = (1-t^{-\alpha})\mathbf{w}_t + t^{-a}\mathbf{v}_t, \quad (25) \end{aligned}$$

where $\alpha = \frac{1}{4}$ and $F_t(\mathbf{w})$ is computed as:

$$\begin{aligned} & F_t(\mathbf{w}) = \frac{1}{t+1}\sum_{i=0}^t \hat{l}_i(\mathbf{w}) \\ & = \frac{1}{t+1}\sum_{i=0}^t(\nabla l_i(\mathbf{w}_i)^T\mathbf{w} + \sigma_i\|\mathbf{w}-\mathbf{w}_0\|_2^2) \\ & = \frac{1}{t+1}\sum_{i=0}^t\Big((\mathbf{w}_i^T\mathbf{x}_i - r_i - \gamma\mathbf{w}_i^T\mathbf{x}_{i+1})(\mathbf{x}_i - \gamma\mathbf{x}_{i+1})^T\mathbf{w} \\ & \quad + \sigma_i\|\mathbf{w}-\mathbf{w}_0\|_2^2\Big), \end{aligned}$$

where $\sigma_t = (L/D)t^{-1/4}$, $\mathbf{w}_0$ is the initialization.

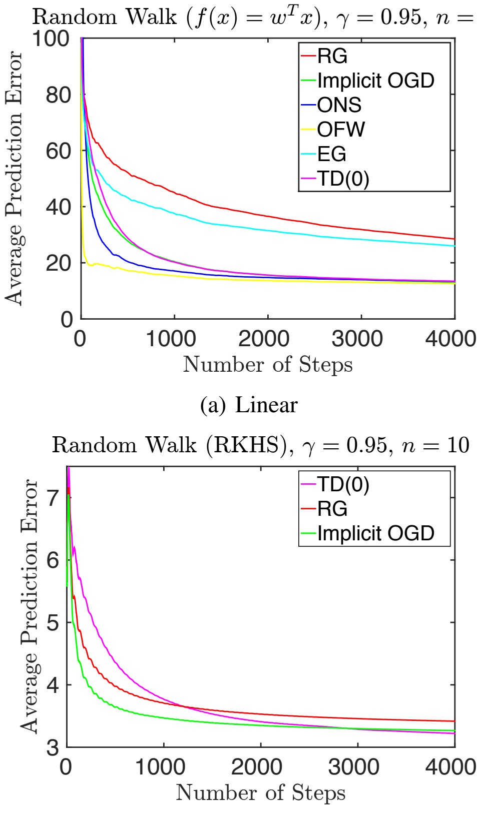
The online stability of OFW can be shown easily:

$$\begin{aligned} & \frac{1}{T}\sum(\mathbf{w}_t^T\mathbf{x}_{t+1} - \mathbf{w}_{t+1}^T\mathbf{x}_{t+1})^2 \\ & \leq \frac{1}{T}X^2\sum\|\mathbf{w}_t - \mathbf{w}_{t+1}\|_2^2 \\ & \leq \frac{1}{T}X^2\sum t^{-2\alpha}\|\mathbf{w}_t - \mathbf{v}_t\|_2^2 \\ & \leq \frac{1}{T}X^2D^2\sum t^{-2\alpha} = 0, \;\; T\rightarrow\infty. \end{aligned}$$

The first inequality comes from Cauchy-Schwartz inequality and the assumption that $\|\mathbf{x}\|_2 \leq X$. The last equality follows from the fact that $2\alpha > 0$, and $\frac{1}{T}\sum_{t=1}^T t^{-\xi} = 0$, when $\xi > 0$ and $T\rightarrow\infty$.

Note that the output of the OFW $\mathbf{w}_t$ is sparse when $\mathcal{W}$ is defined as $\mathcal{W} = \{\mathbf{w} : \|\mathbf{w}\|_1 \leq W'\}$ for some $W' \in \mathbb{R}^+$ and $\mathbf{w}_0$ is initialized to the origin or any corner point of $\mathcal{W}$. This is because the output $\mathbf{v}_t$ of Eq. 25 will be always one of the corner points of $\mathcal{W}$ and $\mathbf{w}_t$ hence is a linear combination of corner points $\{\mathbf{w}_0, \mathbf{v}_0..., \mathbf{v}_{t-1}\}$, leading to the

[2] Usually, when $\mathcal{W}$ is defined as $\{\mathbf{w} : \|\mathbf{w}\|_2 \leq W\}$, an L_2 projection to such $\mathcal{W}$ is easy and can be implemented in $O(n)$. The same for L_2 projection in RKHS when $\mathcal{F} = \{f : \|f\| \leq F\}$.



(a) Linear

(b) RKHS

Figure 1: Convergence of prediction error for Random Walk with linear function approximation (top) and RKHS (bottom).

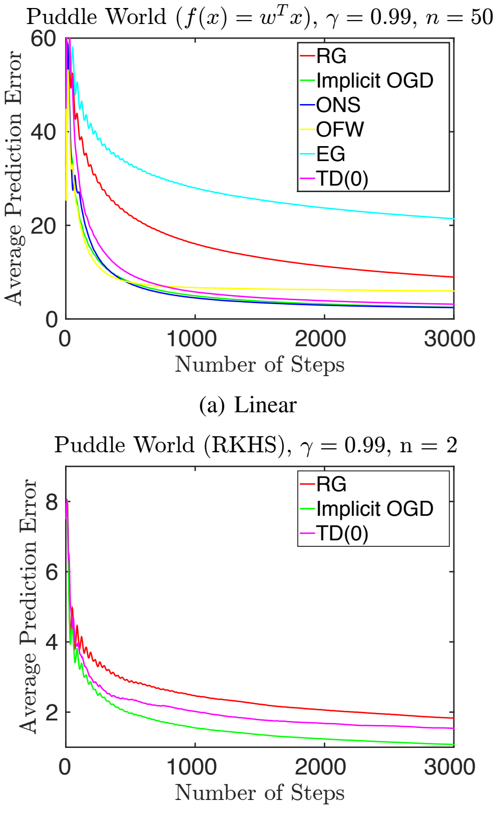
fact that $\mathbf{w}_t$ can have at most $(t+1)$ non-zero entries. Also, if $\mathcal{W} = \{\mathbf{w} : \|\mathbf{w}\|_1 \leq W'\}$, Eq. 25 can be implemented in $O(n)$ as follows: (1) find the entry i in $\nabla F_t(\mathbf{w}_t)$ that has the maximum absolute value; (2) set $\mathbf{v}_t^i$ (the i'th entry) to be $-sign(\nabla F_t(\mathbf{w}_t)^i)W'$, and all other entries in $\mathbf{v}_t$ to zero.

When $\mathcal{W} = \{\mathbf{w} : \|\mathbf{w}\|_2 \leq W\}$, we cannot achieve sparsity, but Eq. 25 can still be implemented in $O(n)$ by setting $\mathbf{v}_t = \frac{-\nabla F_t(\mathbf{w}_t)}{\|\nabla F_t(\mathbf{w}_t)\|_2} W$.

5 EXPERIMENTS

We applied these algorithms to three simulated policy evaluation problems: (1) the Random Walk problem with a ring chain, which is a variant of the Hall problem introduced by Baird (1995), (2) PuddleWorld adopted from Sutton and Barto (1998) and (3) Helicopter Hover using the simulator from Coates et al. (2008). We also compare the above algorithms to standard $\texttt{TD}(0)$ (Sutton and Barto, 1998).

Random Walk The state space of the Random Walk task has $N = 50$ states. The states are linked together as a ring without any terminal states. At time step t, from any state on the ring, the transition probability of moving clockwise is $0.5/\sqrt{t}$ and the probability of moving counterclockwise is $(1 - 0.5/\sqrt{t})$. Note that the transition probability changes over time. We randomly generate a feature vector $\mathbf{x} \in \mathbb{R}^{10}$ and assign it to a state. All rewards are uniformly sampled from $[-3, 3]$. In this problem, we tested both RKHS (Fig. 1(b)) and linear function approximation (Fig. 1(a)). For RKHS, we use a RBF Kernel with bandwidth 0.2, qualitatively chosen for the best performance in terms of prediction error.

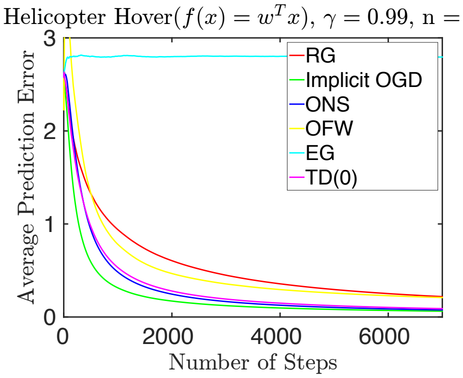


(a) Linear

(b) RKHS

Figure 2: Convergence of prediction error for Puddle World with linear function approximation (top) and RKHS (bottom).

PuddleWorld In the PuddleWorld scenario, the state space is a unit square with "puddles" and the agent's state is represented by its x and y coordinates. In each episode, the agent's starting state is uniformly sampled in the region $[0, 0.2] \times [0, 0.2]$. The policy at each time step selects to go north or east with probability 0.5 each. For each step, the reward is -1 if the agent does not step in a puddle, and the reward decreases quadratically as the agent steps into the puddles. The terminal region is defined as $x + y \geq 1.9$ (upper right corner of the square), which has reward 0. For linear function approximation, we used 50 RBF features ($\mathbf{x} \in \mathbb{R}^{50}$) of bandwidth 0.2, whose centers were uniformly distributed in the state space (Fig. 2(a)). For RKHS, we again use a RBF kernel bandwidth width 0.2 (Fig. 2(b)).



(a) Linear

Figure 3: Convergence of prediction error for Helicopter Hover with linear function approximation

Helicopter Hover The helicopter simulator has a continuous 21-dimensional state space and a continuous 4-dimensional control. The reward is equal to the negative of the quadratic deviation to the targeted hover state. To generate sequence of states, we apply an LQR controller using linearized dynamics around the target hover state. We additionally corrupt the dynamics simulation with noise sampled from a Gaussian distribution. We used a degree-two polynomial feature that maps an original 21-dimensional state to a feature vector $\mathbf{x} \in \mathbb{R}^{253}$ (Fig. 3(a)) and attempt to predict the long-term cost-to-go.

Analysis of results We fixed $\texttt{TD}(0)$'s step-size but a wide range of step-sizes were tried, and the best choice in terms of prediction error was used for $\texttt{TD}(0)$. For $\texttt{RG}$, implicit OGD, and EG, we set the step-size to $c/\sqrt{t}$, where c is a constant. We also tried a range of c and chose the one that leads to the best performance. For all algorithms, we provided the same random initialization. All the results are computed by averaging over 100 random trials. As we can see from Fig. 3, ONS and implicit OGD give good convergence speed in general. Implicit OGD performed well with both RKHS and linear function approximation. Throughout the experiments, we found that implicit OGD was able to use a larger c to speed up convergence while still maintaining good stability. Surprisingly, our experimental results clearly show that our approaches have the possibility to achieve smaller prediction error than $\texttt{TD}(0)$ (e.g., Fig. 2(b), bottom). This runs counter to the fact that the upper bound of prediction error provided by our analysis in Sec. 3 is looser than the upper bound of prediction error of $\texttt{TD}(0)$ from both Li (2008) and Schapire and Warmuth (1996). Though our analysis is more general, further investigation is needed to tighten the worst-case bounds on our approach.

6 CONCLUSION

We established a general connection between the worst-case prediction of long term reward and Bellman errors for stable prediction algorithms. We showed that together with this online stability condition, **any** no-regret online learning algorithm optimizing Bellman errors ensures small prediction errors. The stability condition is weak enough such that most popular no-regret online algorithms satisfy it. Our approach then suggests and provides soundness guarantees for online prediction of long-term reward using a broad new family of algorithms, including Online $\texttt{BE}$ Newton Step, Online $\texttt{BE}$ Frank Wolf, Implicit $\texttt{BE}$ online learning (implicit gradient descent). The analysis itself can be applied to more general function space of hypotheses including Reproducing Kernel Hilbert Space representations and even to discrete hypothesis classes (i.e. trees). However we also want to point out that while our setting is very general, one might expect that in strongly non-Markovian situations there may fail to be a good predictor—e.g., no linear predictor using only the features of x_t can do a good job. In that sense our theorem in this paper is relative—essentially temporally coherent predictions (in the sense of small Bellman error) imply doing nearly as well as can be done at long term prediction: whether that is actually good performance depends on the quality of both features and hypothesis class, but not on any probabilistic assumptions.

7 DISCUSSION

Although our analysis provides broad and sound generalizations of $\texttt{RG}$, it does not provide guarantees on what we believe are the natural generalization of $\texttt{TD}(0)$ or its variants as online algorithms on a sequence of temporal difference loss functions ($\texttt{TD}$-loss) which is defined as:

$$\tilde{l}_t(f) = (f(\mathbf{x}_t) - r_t - \gamma f_t(\mathbf{x}_{t+1}))^2. \tag{26}$$

Note that the difference between $\tilde{l}_t$ and l_t is the subsript on the second predictor. The reason that we call it $\texttt{TD}$-loss is that when $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ is linear, applying OGD to $\tilde{l}_t(\mathbf{w})$ with respect $\mathbf{w}$ exactly reveals the update step of $\texttt{TD}(0)$. By properly choosing step-size ($\mu = O(1/\sqrt{T})$), OGD is no-regret on the $\texttt{TD}$-loss functions $\{\tilde{l}_t(f)\}$. Similar to our analysis of Bellman error algorithms, we believe that it is possible that no-regret property on $\texttt{TD}$-loss functions and stability condition of online algorithms together could lead us to similar predictive guarantees as shown in Theorem 3.3. In fact our empirical results (included in Appendix) suggested that such approaches are both sound and may outperform Bellman Residual methods. We leave it as future work to establish regret bounds for temporal difference minimizing online algorithms.

8 ACKNOWLEDGEMENTS

This work is supported by the ONR MURI grant N00014-09-1-1052, Reasoning in Reduced Information Spaces. We gratefully thank Arun Venkatraman for valuable discussions.

References

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. *Proceedings of the 12th International Conference on Machine Learning (ICML-95)*, pages 30–37, 1995.

Adam Coates, Pieter Abbeel, and Andrew Y Ng. Learning for control from multiple demonstrations. *Proceedings of the 25th international conference on Machine learning - ICML '08*, pages 144–151, 2008.

Yaakov Engel, Shie Mannor, and Ron Meir. Reinforcement learning with Gaussian processes. *Proceedings of the 22nd international conference on Machine learning*, 2005.

Elad Hazan and Satyen Kale. Projection-free Online Learning. *29th International Conference on Machine Learning (ICML 2012)*, pages 521–528, 2012.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Proceedings of the 19th annual conference on Computational Learning Theory (COLT)*, pages 169–192, 2006.

Brian Kulis, Peter L Bartlett, Bartlett Eecs, and Berkeley Edu. Implicit Online Learning. *Proceedings of the 27th international conference on Machine learning (ICML)*, pages 575–582, 2010.

Lihong Li. A worst-case comparison between temporal difference and residual gradient with linear function approximation. *Proceedings of the 25th international conference on Machine learning - ICML '08*, pages 560–567, 2008.

Doina Precup and Richard S. Sutton. Exponentiated Gradient Methods for Reinforcement Learning. In *Proceedings of the 14th International Conference on Machine Learning (ICML)*, 1997.

M Robards, Peter Sunehag, Scott Sanner, and B Marthi. Sparse Kernel-SARSA(lambda) with an Eligibility Trace. *ECML PKDD*, 2011.

Stephane Ross and J. Andrew Bagnell. Stability Conditions for Online Learnability. *arXiv:1108.3154*, 2011.

Ankan Saha, Prateek Jain, and Ambuj Tewari. The Interplay Between Stability and Regret in Online Learning. *arXiv preprint arXiv:1211.6158*, pages 1–19, 2012.

Robert E. Schapire and Manfred K. Warmuth. On the worst-case analysis of temporal-difference learning algorithms. *Machine Learning*, 22(1):95–121, 1996. ISSN 0885-6125. doi: 10.1007/BF00114725.

Bruno Scherrer. Should one compute the Temporal Difference fix point or minimize the Bellman Residual? The unified oblique projection view. *International Conference on Machine Learning (ICML 2010)*, 2010.

Ralf Schoknecht and Artur Merke. TD(0) Converges Provably Faster than the Residual Gradient Algorithm. *International Conference on Machine Learning (ICML 2003)*, pages 680–687, 2003.

Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press Cambridge, MA, USA, 2001.

Shai Shalev-Shwartz. Online Learning and Online Convex Optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, 1998.

Aviv Tamar, Panos Toulis, Shie Mannor, and Edoardo M. Airoldi. Implicit Temporal Differences. *arXiv:1412.6734*, pages 1–6, 2014.

Martin Zinkevich. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In *International Conference on Machine Learning (ICML 2003)*, pages 421–422, 2003.