Machine Learning, 51, 23–50, 2003

Relative Loss Bounds for Temporal-Difference Learning

JÜRGEN FORSTER forster@lmi.ruhr-uni-bochum.de
Lehrstuhl Mathematik & Informatik, Fakultät für Mathematik, Ruhr-Universität Bochum, 44780 Bochum, Germany

MANFRED K. WARMUTH manfred@cse.ucsc.edu
Computer Science Department, University of California, Santa Cruz, CA 95064, USA

Editor: Peter Bartlett

Abstract. Foster and Vovk proved relative loss bounds for linear regression where the total loss of the on-line algorithm minus the total loss of the best linear predictor (chosen in hindsight) grows logarithmically with the number of trials. We give similar bounds for temporal-difference learning. Learning takes place in a sequence of trials where the learner tries to predict discounted sums of future reinforcement signals. The quality of the predictions is measured with the square loss and we bound the total loss of the on-line algorithm minus the total loss of the best linear predictor for the whole sequence of trials. Again the difference of the losses is logarithmic in the number of trials. The bounds hold for an arbitrary (worst-case) sequence of examples. We also give a bound on the expected difference for the case when the instances are chosen from an unknown distribution. For linear regression a corresponding lower bound shows that this expected bound cannot be improved substantially.

Keywords: machine learning, temporal-difference learning, on-line learning, relative loss bounds

1. Introduction

In reinforcement learning the learner interacts with its environment by iteratively observing the state of the environment and choosing actions according to a policy. Reinforcement signals are a short-term estimation of the current state of the environment. The learner's task is to find a policy such that the long-term outcome, which is a numerical function of the reinforcement signals, is maximized. The outcome is typically defined to be a (discounted) sum of the future reinforcement signals. This means that the outcome values cannot be observed directly by the learner.

An important subproblem of reinforcement learning is to estimate the value function for a given policy. The value function of a policy assigns to each state of the environment a measure of the long-term performance given that the system starts in that state and the learner follows the policy to choose its actions. In this paper we consider the following model of value function approximation by linear functions: Learning takes place in a sequence of trials where the learner tries to predict discounted sums of future reinforcement signals. The quality of the predictions is measured with the square loss and we bound the total loss of the on-line algorithm minus the total loss of the best linear predictor for the whole sequence of trials. Though relative loss bounds of this kind are often hard to prove, they have the

advantage over more traditional bounds that they hold for all sequences of states. Hence any dependence between successive states does not break the analysis.

More formally, consider the following model of temporal-difference learning: Learning proceeds in a sequence of *trials* $t = 1, 2, \ldots$, where at trial t,

- the learner receives an *instance vector* $\boldsymbol{x}_t \in \mathbb{R}^n$ that contains the available information about the state of the environment,
- the learner makes a *prediction* $\hat{y}_t \in \mathbb{R}$,
- the learner receives a *reinforcement signal* $r_t \in \mathbb{R}$.

A pair $(\boldsymbol{x}_t, r_t)$ is called an *example*. The learner tries to predict the *outcomes* $y_t \in \mathbb{R}$. For a fixed *discount rate parameter* $\gamma \in [0, 1)$, y_t is the discounted sum

$$y_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots = \sum_{s=t}^{\infty} \gamma^{s-t} r_s \tag{1.1}$$

of the future reinforcement signals.[1] For example, if r_t is the profit of a company in month t, then y_t can be interpreted as an approximation of the company's worth at time t. The discounted sum takes into account that profits in the distant future are less important than short term profits. Note that the outcome y_t as defined in (1.1) is well-defined if the reinforcement signals are bounded. In an episodic setting one can also define the outcomes y_t as finite discounted sums. This is discussed briefly in Section 7.

A strategy that chooses predictions for the learner is called an *on-line learning algorithm*. The quality of a prediction is measured with the *square loss*: The loss of the learner at trial t is $(y_t - \hat{y}_t)^2$, and the loss of the learner at trials 1 through T is $\sum_{t=1}^{T} (y_t - \hat{y}_t)^2$.

We want to compare the loss of the learner against the losses of linear functions. A linear function is represented by a weight vector $\boldsymbol{w} \in \mathbb{R}^n$, and the loss of $\boldsymbol{w}$ at trial t is $(y_t - \boldsymbol{w} \cdot \boldsymbol{x}_t)^2$. Ideally we want to bound the additional loss of the learner over the loss of the best linear predictor for arbitrary sequences of examples, i.e. we want to bound

$$\sum_{t=1}^{T} (y_t - \hat{y}_t)^2 - \inf_{\boldsymbol{w} \in \mathbb{R}^n} \left(\sum_{t=1}^{T} (y_t - \boldsymbol{w} \cdot \boldsymbol{x}_t)^2 \right) \tag{1.2}$$

for arbitrary T and arbitrary sequences of examples $(\boldsymbol{x}_1, r_1), (\boldsymbol{x}_2, r_2), \ldots$ The first sum in (1.2) is the total loss of the learner at trials 1 through T. The argument of the infimum is the total loss of the linear function $\boldsymbol{w}$ at trials 1 through T. Thus (1.2) is the additional total loss of the learner over the total loss of the best linear function.

Following Vovk (1997) we also examine the more general problem of bounding

$$\sum_{t=1}^{T} (y_t - \hat{y}_t)^2 - \inf_{\boldsymbol{w} \in \mathbb{R}^n} \left(\sum_{t=1}^{T} (y_t - \boldsymbol{w} \cdot \boldsymbol{x}_t)^2 + a \|\boldsymbol{w}\|^2 \right) \tag{1.3}$$

for a fixed constant $a \geq 0$. Here $a\|\boldsymbol{w}\|^2$ is a measure of the complexity of $\boldsymbol{w}$, i.e. the infimum in (1.3) includes a charge for the complexity of the linear function. For larger values of a it is obviously easier to show bounds on (1.3).

Bounds on (1.2) or (1.3) that hold for arbitrary sequences of examples are called *relative loss bounds*. Relative loss bounds for the temporal-difference learning setting were first shown by Schapire and Warmuth (1996). An overview of their results is given in Section 4. They also show how algorithms that minimize (1.2) can be used for value function approximation of Markov processes. This is an important problem in reinforcement learning which is also called policy evaluation. The Markov processes can have continuously many states which are represented by real vectors. If one wants to predict state values then our instances $\boldsymbol{x}_t$ correspond to states of the environment, if action values are predicted then an instance corresponds to both a state of the environment and an action of the agent. For an introduction to reinforcement learning see Sutton and Barto (1998).

The paper is organized as follows. We discuss previously known relative loss bounds for linear regression and for temporal-difference learning in Sections 3 and 4. In Section 5, we propose a new second order learning algorithm for temporal-difference learning (the TLS algorithm), and we prove relative loss bounds for this algorithm in Section 6. In Section 7, we adapt the TLS algorithm to the episodic case, where the trials are divided into episodes and where an outcome is a discounted sum of the future reinforcement signals from the same episode. We discuss previous second order algorithms for temporal-difference learning in Section 8, and give lower bounds on the relative loss in Section 9.

2. Notation and preliminaries

For $n \in \mathbb{N}$, $\mathbb{R}^n$ is the set of n-dimensional real vectors. For $m, n \in \mathbb{N}$, $\mathbb{R}^{m \times n}$ is the set of real matrices with m rows and n columns. In this paper vectors $\boldsymbol{x} \in \mathbb{R}^n$ are column vectors and $\boldsymbol{x}'$ denotes the transpose of $\boldsymbol{x}$. The scalar product of two vectors $\boldsymbol{w}, \boldsymbol{x} \in \mathbb{R}^n$ is $\boldsymbol{w} \cdot \boldsymbol{x} = \boldsymbol{w}'\boldsymbol{x}$, and the Euclidean norm of a vector $\boldsymbol{x} \in \mathbb{R}^n$ is $\|\boldsymbol{x}\| = (\boldsymbol{x} \cdot \boldsymbol{x})^{\frac{1}{2}}$.

We recall some basic facts about positive (semi-)definite matrices:

- A symmetric matrix $A \in \mathbb{R}^{n \times n}$ is called *positive definite* if $\boldsymbol{x}' A \boldsymbol{x} > 0$ holds for all vectors $\boldsymbol{x} \in \mathbb{R}^n \backslash \{\boldsymbol{0}\}$.
- A symmetric matrix $A \in \mathbb{R}^{n \times n}$ is called *positive semi-definite* if $\boldsymbol{x}' A \boldsymbol{x} \geq 0$ for all $\boldsymbol{x} \in \mathbb{R}^n$.
- The sum of two positive semi-definite matrices is again positive semi-definite. The sum of a positive semi-definite matrix and a positive definite matrix is positive definite.
- Every positive definite matrix is invertible.
- For matrices $A, B \in \mathbb{R}^{n \times n}$ we write $A \leq B$ if $B - A$ is positive semi-definite. In this case $\boldsymbol{x}' A \boldsymbol{x} \leq \boldsymbol{x}' B \boldsymbol{x}$ for all vectors $\boldsymbol{x} \in \mathbb{R}^n$.
- The Sherman-Morrison formula (see Press et al., 1989)

$$(A + \boldsymbol{x}\boldsymbol{x}')^{-1} = A^{-1} - \frac{(A^{-1}\boldsymbol{x})(A^{-1}\boldsymbol{x})'}{1 + \boldsymbol{x}' A^{-1} \boldsymbol{x}} \tag{2.1}$$

  holds for every positive definite matrix $A \in \mathbb{R}^{n \times n}$ and every vector $\boldsymbol{x} \in \mathbb{R}^n$.

For example, the unit matrix $I \in \mathbb{R}^{n \times n}$ is positive definite and for every vector $\boldsymbol{x} \in \mathbb{R}^n$ the matrix $\boldsymbol{x}\boldsymbol{x}' \in \mathbb{R}^{n \times n}$ is positive semi-definite.

To find a vector $\boldsymbol{w} \in \mathbb{R}^n$ that minimizes the term

$$\sum_{s=1}^{t}(y_s - \boldsymbol{w} \cdot \boldsymbol{x}_s)^2 + a\|\boldsymbol{w}\|^2 \tag{2.2}$$

that appears in the relative loss (1.3) we define

$$\boldsymbol{b}_t := \sum_{s=1}^{t} y_s \boldsymbol{x}_s \overset{(1.1)}{=} \sum_{s=1}^{t}\left(\sum_{k=s}^{\infty}\gamma^{k-s} r_k\right)\boldsymbol{x}_s \in \mathbb{R}^n, \tag{2.3}$$

$$A_t := aI + \sum_{s=1}^{t}\boldsymbol{x}_s\boldsymbol{x}_s' \in \mathbb{R}^{n\times n}. \tag{2.4}$$

Because (2.2) is convex and continuously differentiable in $\boldsymbol{w}$, it is minimal if and only if its gradient is zero, i.e. if and only if $A_t\boldsymbol{w} = \boldsymbol{b}_t$. If $a > 0$, then A_t is invertible and $\boldsymbol{w} = A_t^{-1}\boldsymbol{b}_t$ is the unique vector that minimizes (2.2). If $a = 0$ then A_t might not be invertible and the equation $A_t\boldsymbol{w} = \boldsymbol{b}_t$ might have multiple solutions. The solution with the smallest Euclidean norm is $\boldsymbol{w} = A_t^+\boldsymbol{b}_t$, where A_t^+ is the *pseudoinverse* of A_t. For the definition of the pseudoinverse of a matrix see, e.g., Rektorys (1994). There it is also shown how the pseudoinverse of a matrix can be computed with the singular value decomposition. We give a number of properties of the pseudoinverse A_t^+ in Appendix A.

If $a > 0$, then applying the Sherman-Morrison formula (2.1) to $A_t = A_{t-1} + \boldsymbol{x}_t\boldsymbol{x}_t'$ shows that

$$A_t^{-1} = A_{t-1}^{-1} - \frac{\left(A_{t-1}^{-1}\boldsymbol{x}_t\right)\left(A_{t-1}^{-1}\boldsymbol{x}_t\right)'}{1 + \boldsymbol{x}_t' A_{t-1}^{-1}\boldsymbol{x}_t}. \tag{2.5}$$

3. Known relative loss bounds for linear regression

First note that linear regression is a special case of our setup since when $\gamma = 0$ then by (1.1) $y_t = r_t$ for all t. The standard algorithm for linear regression is the ridge regression algorithm which predicts with $\hat{y}_t = \boldsymbol{b}_{t-1}' A_{t-1}^{-1}\boldsymbol{x}_t$ at trial t. Relative loss bounds for this algorithm (similar to the bounds given in the below two theorems) have been proven in Foster (1991), Vovk (1997) and Azoury and Warmuth (2001). The bounds obtained for ridge regression are weaker than the ones proven for a new algorithm developed by Vovk. We will give a simple motivation of this algorithm and then discuss the relative loss bounds that were proven for it.

We have seen in Section 2 that the best linear functions for trials 1 through t (i.e. the linear function that minimizes (2.2)) would make the prediction $\boldsymbol{b}_t' A_t^+\boldsymbol{x}_t$ at trial t. (The pseudoinverse A_t^+ of A_t reduces to the inverse A_t^{-1} if A_t is invertible.) Note that via $\boldsymbol{b}_t$ and A_t this prediction depends on the examples $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_t, y_t)$. However, only the examples $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{t-1}, y_{t-1})$ and the instance $\boldsymbol{x}_t$ are known to the learner when it makes the prediction for trial t. If we set the unknown outcome y_t to zero we get the prediction $\boldsymbol{b}_{t-1}' A_t^+\boldsymbol{x}_t$. This prediction was introduced in Vovk (1997) using a different

motivation. The above motivation follows Azoury and Warmuth (2001). Forster (1999) gives an alternate game theoretic motivation. Vovk proved the following bound on (1.3) for his prediction algorithm.

Theorem 3.1. *Consider linear regression* ($\gamma = 0$) *with* $a > 0$. *Let* $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_T, y_T)$ *be any sequence of examples in* $\mathbb{R}^n \times [-Y, Y]$. *Then with the predictions* $\hat{y}_t = \boldsymbol{b}'_{t-1} A_t^{-1} \boldsymbol{x}_t$,

$$\begin{aligned}
&\sum_{t=1}^{T} (y_t - \hat{y}_t)^2 - \inf_{\boldsymbol{w} \in \mathbb{R}^n} \left(\sum_{t=1}^{T} (y_t - \boldsymbol{w} \cdot \boldsymbol{x}_t)^2 + a \|\boldsymbol{w}\|^2 \right) \\
&\quad \le Y^2 \ln \det \left(\frac{1}{a} A_T \right) \le Y^2 \ln \prod_{i=1}^{n} \left(1 + \frac{1}{a} \sum_{t=1}^{T} x_{t,i}^2 \right) \\
&\quad \le n Y^2 \ln \left(1 + \frac{TX^2}{an} \right),
\end{aligned}$$

where $x_{t,i}$ *is the* i*-th component of the vector* $\boldsymbol{x}_t$ *and where* $X = \max_{t \in \{1,\ldots,T\}} \|\boldsymbol{x}_t\|$.

In Vovk's version of Theorem 3.1 the term $\frac{X^2}{n}$ is replaced by the larger term X_∞^2, where X_∞ is a bound on the supremum norms of the instances $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$. The last inequality in Theorem 3.1 follows from

$$\prod_{i=1}^{n} \left(1 + \frac{1}{a} \sum_{t=1}^{T} \boldsymbol{x}_{t,i}^2 \right) \le \left(1 + \frac{1}{an} \sum_{t=1}^{T} \underbrace{\sum_{i=1}^{n} \boldsymbol{x}_{t,i}^2}_{= \|\boldsymbol{x}_t\|^2} \right)^n \le \left(1 + \frac{TX^2}{an} \right)^n,$$

where the first inequality holds because the geometric mean is always smaller than the arithmetic mean.

Azoury and Warmuth (2001) and Forster (1999) give an alternate proof of Theorem 3.1. They provide some intermediate steps that we need in this paper. We also extend their bounds for $a > 0$ to the case $a = 0$ by a limiting argument.

Theorem 3.2. *Consider linear regression* ($\gamma = 0$). *Let* $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_T, y_T)$ *be any sequence of examples in* $\mathbb{R}^n \times \mathbb{R}$.

(i) *If* $a \ge 0$, *then with the predictions* $\hat{y}_t = \boldsymbol{b}'_{t-1} A_t^{+} \boldsymbol{x}_t$,

$$\begin{aligned}
&\sum_{t=1}^{T} (y_t - \hat{y}_t)^2 - \inf_{\boldsymbol{w} \in \mathbb{R}^n} \left(\sum_{t=1}^{T} (y_t - \boldsymbol{w} \cdot \boldsymbol{x}_t)^2 + a \|\boldsymbol{w}\|^2 \right) \\
&\quad = \sum_{t=1}^{T} y_t^2 \boldsymbol{x}'_t A_t^{+} \boldsymbol{x}_t - \sum_{t=1}^{T} (\boldsymbol{b}'_{t-1} A_t^{+} \boldsymbol{x}_t)^2 \, \boldsymbol{x}'_t A_{t-1}^{+} \boldsymbol{x}_t \\
&\quad \le \sum_{t=1}^{T} y_t^2 \boldsymbol{x}'_t A_t^{+} \boldsymbol{x}_t.
\end{aligned}$$

(ii) *If* $a > 0$, *then*

$$\sum_{t=1}^{T} \boldsymbol{x}_t' A_t^{-1} \boldsymbol{x}_t \leq \ln \det\left(\frac{1}{a} A_T\right)$$

for all vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T \in \mathbb{R}^n$.

Proof: We only have to show that the equality in (i) also holds for the case $a = 0$. We do this by showing that both sides of the equality are continuous in $a \in [0, \infty)$. Because of Lemma A.2 we only have to check that for $t \in \{1, \ldots, T\}$ the term

$$(\boldsymbol{b}_{t-1}' A_t^{+} \boldsymbol{x}_t)^2 \, \boldsymbol{x}_t' A_{t-1}^{+} \boldsymbol{x}_t \tag{3.1}$$

is continuous in $a \in [0, \infty)$. If $\boldsymbol{x}_t \in \text{span}\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}\}$, this again follows from Lemma A.2. Otherwise $\boldsymbol{x}_t \notin \text{span}\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}\}$. Then by Lemma A.3, the expression (3.1) is zero for $a = 0$. We have to show that (3.1) converges to zero for $a \searrow 0$. For $a > 0$, we can rewrite (3.1) by applying (2.5):

$$\left(\boldsymbol{b}_{t-1}' A_{t-1}^{-1} \boldsymbol{x}_t\right)^2 \frac{\boldsymbol{x}_t' A_{t-1}^{-1} \boldsymbol{x}_t}{\left(1 + \boldsymbol{x}_t' A_{t-1}^{-1} \boldsymbol{x}_t\right)^2}. \tag{3.2}$$

The factor $(\boldsymbol{b}_{t-1}' A_{t-1}^{-1} \boldsymbol{x}_t)^2$ converges because of Lemma A.2. Because of Lemma A.4, the term $\frac{\boldsymbol{x}_t' A_{t-1}^{-1} \boldsymbol{x}_t}{(1+\boldsymbol{x}_t' A_{t-1}^{-1} \boldsymbol{x}_t)^2}$ goes to zero as $a \searrow 0$. Together this shows that (3.1) indeed goes to zero as $a \searrow 0$. □

The learning algorithm of Theorem 3.1 and Theorem 3.2 is a *second order* algorithm in that it uses second derivatives. Relative loss bounds for linear regression that are logarithmic in the number of trials T have only been shown for second order algorithms. There is a simpler *first order* algorithm called the Widrow-Hoff or Least Mean Square algorithm (Widrow & Stearns, 1985). This algorithm maintains a weight vector $\boldsymbol{w}_t \in \mathbb{R}^n$ and predicts with $\hat{y}_t = \boldsymbol{w}_t \cdot \boldsymbol{x}_t$. The weight vector is updated by gradient descent. That is, $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t(\boldsymbol{w}_t \cdot \boldsymbol{x}_t - y_t)\boldsymbol{x}_t$, for some *learning rates* $\eta_t > 0$.

A method for setting the learning rates for the purpose of obtaining good relative loss bounds is given in Cesa-Bianchi, Long, and Warmuth (1996) and Kivinen and Warmuth (1997). In this method the learner needs to know an upper bound X on the Euclidean norms of the instances and needs to know parameters W, K such that there is a vector $\boldsymbol{w} \in \mathbb{R}^n$ with norm $\|\boldsymbol{w}\| \leq W$ and loss $\sum_{t=1}^{T}(y_t - \boldsymbol{w} \cdot \boldsymbol{x}_t)^2 \leq K$. For any such vector $\boldsymbol{w}$ the bound

$$\sum_{t=1}^{T}(y_t - \hat{y}_t)^2 \leq \sum_{t=1}^{T}(y_t - \boldsymbol{w} \cdot \boldsymbol{x}_t)^2 + 2\sqrt{K}WX + \|\boldsymbol{w}\|^2 X^2 \tag{3.3}$$

holds. As noted by Vovk (1997) this bound is incomparable to the bounds of Theorem 3.1 (see also the next section). The bound (3.3) also holds for the ridge regression algorithm (Hassibi, Kivinen, & Warmuth, 1995). There the parameter a is set depending on W and K. We believe that with a proper tuning of a such bounds also hold for Vovk's algorithm.

4. Known relative loss bounds for temporal-difference learning

In the previous section we considered the case of linear regression where $\gamma = 0$. Now we want to talk about the case of real temporal-difference learning, where the discount rate parameter γ is larger than zero, i.e. the outcomes can depend on all of the future reinforcement signals.

Schapire and Warmuth (1996) have given a number of different relative loss bounds for the learning algorithm TD*(λ) (λ ∈ [0, 1]). The first order algorithm TD*(λ) is essentially a generalization of the Widrow-Hoff algorithm. It is a slight modification of the learning algorithm TD(λ) proposed by Sutton (1988). Schapire and Warmuth show that the loss of TD*(1) with a specific setting of its learning rate is

$$\sum_{t=1}^{T}(y_t - \hat{y}_t)^2 \leq \sum_{t=1}^{T}(y_t - \boldsymbol{w} \cdot \boldsymbol{x}_t)^2 + 2\sqrt{K}WXc_\gamma + \|\boldsymbol{w}\|^2 X^2 c_\gamma^2 \tag{4.1}$$

(where $c_\gamma = \frac{1+\gamma}{1-\gamma}$) and that the loss of the algorithm TD*(0) with a specific setting of its learning rate is

$$\sum_{t=1}^{T}(y_t - \hat{y}_t)^2 \leq \frac{\sum_{t=1}^{T}(y_t - \boldsymbol{w} \cdot \boldsymbol{x}_t)^2 + 2\sqrt{K}WX + \|\boldsymbol{w}\|^2 X^2}{1 - \gamma^2} \tag{4.2}$$

for every vector $\boldsymbol{w} \in \mathbb{R}^n$ with $\|\boldsymbol{w}\| \leq W$ and $\sum_{t=1}^{T}(y_t - \boldsymbol{w} \cdot \boldsymbol{x}_t)^2 \leq K$. The setting of the learning rate depends on an upper bound X on the Euclidean norms of the instances and on W and K. The learner needs to know these parameters in advance.

The loss of the best linear function will often grow linearly in T, e.g. if the examples are corrupted by Gaussian noise. In this case the relative loss bounds in (3.3), (4.1) and (4.2) will grow like $\sqrt{T}$. The second order learning algorithm we propose for temporal-difference learning has the advantage that the relative loss bounds we can prove for it grow only logarithmically in T. Also, our algorithm does not need to know parameters like K and W. However, it needs to know an upper bound Y on the absolute values of the outcomes y_t.

TD*(λ) can be sensitive to the choice of λ. Another advantage of our algorithm is that we do not have to choose a parameter like the λ of the TD*(λ) algorithm.

5. A new second order algorithm for temporal-difference learning

In this section we propose a new algorithm for the temporal-difference learning setting. We call this algorithm the temporal least squares algorithm, or shorter the TLS algorithm.

We assume that the absolute values of the outcomes y_t are bounded by some constant Y, i.e.

$$\forall t\colon |y_t| \leq Y, \tag{5.1}$$

and assume that the bound Y, the discount rate parameter γ, and the parameter a are known to the learner. Knowing Y the learner can "clip" a real number $y \in \mathbb{R}$ using the

function

$$C_Y(y) := \begin{cases} Y, & y \geq Y, \\ y, & y \in [-Y, Y], \\ -Y, & y \leq -Y. \end{cases} \tag{5.2}$$

Note that this "clipping" never increases the loss of the learner: if the outcomes are known to lie in the interval $[-Y, Y]$ the learner should not predict with a number larger than Y or smaller than $-Y$.

In the temporal-difference learning setting we do not know the outcomes $y_1, \ldots, y_{t-1}$ when we need to predict at trial t. So we cannot run Vovk's linear regression algorithm which uses these outcomes in its prediction. The TLS algorithm approximates Vovk's prediction by setting the future reinforcement signals to zero (It also clips the prediction into the range $[-Y, Y]$). We will show that the loss of the TLS algorithm is not much worse than the loss of Vovk's algorithm for which good relative loss bounds are known.

5.1. *Motivation of the TLS algorithm*

The new second-order algorithm for temporal-difference learning is given in Table 1. We call this algorithm the Temporal Least Squares (TLS) algorithm. The motivation for the TLS algorithm is the same as the motivation from Azoury and Warmuth (2001) that we gave for Vovk's prediction for linear regression in Section 3. We will use the equality

$$y_s \overset{(1.1)}{=} \left(\sum_{k=s}^{t-1} \gamma^{k-s} r_k \right) + \gamma^{t-s} y_t \tag{5.3}$$

Table 1. The temporal least squares (TLS) algorithm.

At trial t, the learner knows

- the parameters Y, γ, a,
- the instances $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t$,
- the reinforcement signals $r_1, \ldots, r_{t-1}$.

TLS predicts with

$$\hat{y}_t = C_Y(\tilde{\boldsymbol{b}}'_{t-1} A_t^{-1} \boldsymbol{x}_t),$$

where

$$A_t = aI + \sum_{s=1}^{t} \boldsymbol{x}_s \boldsymbol{x}'_s,$$
$$\tilde{\boldsymbol{b}}_t = \sum_{s=1}^{t} \left(\sum_{k=s}^{t} \gamma^{k-s} r_k \right) \boldsymbol{x}_s,$$

and C_Y given by (5.2) clips the prediction to the interval $[-Y, Y]$.

If A_t is not invertible, then the inverse A_t^{-1} of A_t must be replaced by the pseudoinverse A_t^+.

that holds for all $s \leq t$. This equality means that the outcome of trial s is a discounted sum of the future reinforcement signals of trials $s, \ldots, t-1$ and the outcome of the future trial t. The linear function that minimizes (2.2) would make the prediction

$$\begin{aligned} \boldsymbol{b}_t' A_t^+ \boldsymbol{x}_t &\overset{(2.3)}{=} \left(\sum_{s=1}^{t} y_s \boldsymbol{x}_s \right)' A_t^+ \boldsymbol{x}_t \\ &\overset{(5.3)}{=} \left(\left(\sum_{s=1}^{t} \sum_{k=s}^{t-1} \gamma^{k-s} r_k \boldsymbol{x}_s \right) + y_t \sum_{s=1}^{t} \gamma^{t-s} \boldsymbol{x}_s \right)' A_t^+ \boldsymbol{x}_t \end{aligned} \tag{5.4}$$

at trial t. We set the unknown outcome y_t to zero. If we assume that the outcomes are in a range centered at zero then this is a sensible guess. With this choice of y_t the prediction (5.4) can be rewritten as $\tilde{\boldsymbol{b}}_{t-1}' A_t^+ \boldsymbol{x}_t$, where

$$\tilde{\boldsymbol{b}}_t := \sum_{s=1}^{t} \left(\sum_{k=s}^{t} \gamma^{k-s} r_k \right) \boldsymbol{x}_s. \tag{5.5}$$

The TLS algorithm predicts with $\hat{y}_t = C_Y(\tilde{\boldsymbol{b}}_{t-1}' A_t^+ \boldsymbol{x}_t)$, where the clipping function ensures that the prediction lies in the bounded range $[-Y, Y]$. In the following we will show that the relative loss (1.3) of TLS is at most

$$Y^2 \left(1 + \frac{4\gamma}{1-\gamma} \right) \sum_{t=1}^{T} \boldsymbol{x}_t' A_t^+ \boldsymbol{x}_t$$

and we will use this result to get worst and average case relative loss bounds that are easier to interpret.

5.2. *Implementation of the TLS algorithm*

For the case $a > 0$ a straightforward implementation of the TLS algorithm would need $O(n^3)$ arithmetic operations at each trial to compute the inverse of the matrix A_t. In Table 2 we give an implementation that only needs $O(n^2)$ arithmetic operations per trial. This is achieved by computing the inverse of A_t iteratively using the Sherman-Morrison formula (2.5).

This implementation makes the correct predictions because at the end of each FOR-loop

$$A_{\text{inv}} = A_t^{-1}, \quad \boldsymbol{z} = \sum_{s=1}^{t} \gamma^{t-s} \boldsymbol{x}_s, \quad \tilde{\boldsymbol{b}} = \tilde{\boldsymbol{b}}_t.$$

This follows from the Sherman-Morrison formula (2.5) and from the equality

$$\tilde{\boldsymbol{b}}_t = \tilde{\boldsymbol{b}}_{t-1} + r_t \sum_{s=1}^{t} \gamma^{t-s} \boldsymbol{x}_s.$$

Table 2. Implementation of TLS for $a > 0$.

$A_{\text{inv}} := \frac{1}{a} I \in \mathbb{R}^{n \times n}$
$\tilde{\boldsymbol{b}} := \mathbf{0} \in \mathbb{R}^n$
$z := \mathbf{0} \in \mathbb{R}^n$
FOR $t = 1, 2, 3, \ldots$ DO
  Receive instance vector $\boldsymbol{x}_t \in \mathbb{R}^n$
  $A_{\text{inv}} := A_{\text{inv}} - \dfrac{(A_{\text{inv}}\boldsymbol{x}_t)(A_{\text{inv}}\boldsymbol{x}_t)'}{1 + \boldsymbol{x}_t' A_{\text{inv}} \boldsymbol{x}_t}$
  Predict with $\hat{y}_t = C_Y(\tilde{\boldsymbol{b}}' A_{\text{inv}} \boldsymbol{x}_t) \in \mathbb{R}$
  Receive reinforcement signal $r_t \in \mathbb{R}$
  $z := \gamma z + \boldsymbol{x}_t$
  $\tilde{\boldsymbol{b}} := \tilde{\boldsymbol{b}} + r_t z$
END FOR

6. Relative loss bounds for the TLS algorithm

We start by showing two lemmas. The first is a technical lemma. We use it for proving the second lemma in which we bound the absolute values of the differences between Vovk's prediction $\boldsymbol{b}_{t-1}' A_t^+ \boldsymbol{x}_t$ and the unclipped prediction $\tilde{\boldsymbol{b}}_{t-1}' A_t^+ \boldsymbol{x}_t$ of the TLS algorithm.

Lemma 6.1. *If $a \geq 0$ and $s \leq t$, then for any vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t \in \mathbb{R}^n$:*

$$|\boldsymbol{x}_s' A_t^+ \boldsymbol{x}_t| \leq \frac{1}{2}(\boldsymbol{x}_s' A_s^+ \boldsymbol{x}_s + \boldsymbol{x}_t' A_t^+ \boldsymbol{x}_t).$$

Proof: From

$$0 \leq (\boldsymbol{x}_s \pm \boldsymbol{x}_t)' A_t^+ (\boldsymbol{x}_s \pm \boldsymbol{x}_t) = \boldsymbol{x}_s' A_t^+ \boldsymbol{x}_s + \boldsymbol{x}_t' A_t^+ \boldsymbol{x}_t \pm 2\boldsymbol{x}_s' A_t^+ \boldsymbol{x}_t$$

it follows that

$$2|\boldsymbol{x}_s' A_t^+ \boldsymbol{x}_t| \leq \boldsymbol{x}_s' A_t^+ \boldsymbol{x}_s + \boldsymbol{x}_t' A_t^+ \boldsymbol{x}_t.$$

If $a > 0$, then the pseudoinverses are inverses and the Sherman-Morrison formula (2.5) shows that

$$A_1^{-1} \geq A_2^{-1} \geq A_3^{-1} \geq \cdots$$

Thus $A_t^{-1} \leq A_s^{-1}$ and $\boldsymbol{x}_s' A_t^{-1} \boldsymbol{x}_s \leq \boldsymbol{x}_s' A_s^{-1} \boldsymbol{x}_s$. This proves the lemma for $a > 0$. For the case $a = 0$ we use Lemma A.2 and let a go to 0. □

Lemma 6.2.

$$\sum_{t=1}^{T} |\boldsymbol{b}'_{t-1} A_t^+ \boldsymbol{x}_t - \tilde{\boldsymbol{b}}'_{t-1} A_t^+ \boldsymbol{x}_t| \le Y \frac{\gamma}{1-\gamma} \sum_{t=1}^{T} \boldsymbol{x}'_t A_t^+ \boldsymbol{x}_t.$$

Proof: Note that

$$\boldsymbol{b}_t - \tilde{\boldsymbol{b}}_t \overset{(2.3),(5.5)}{=} \sum_{s=1}^{t} \left(\sum_{k=t+1}^{\infty} \gamma^{k-s} r_k \right) \boldsymbol{x}_s \overset{(1.1)}{=} y_{t+1} \sum_{s=1}^{t} \gamma^{t+1-s} \boldsymbol{x}_s. \tag{6.1}$$

Thus

$$\begin{aligned}
&\sum_{t=1}^{T} |\boldsymbol{b}'_{t-1} A_t^+ \boldsymbol{x}_t - \tilde{\boldsymbol{b}}'_{t-1} A_t^+ \boldsymbol{x}_t| = \sum_{t=1}^{T} |(\boldsymbol{b}_{t-1} - \tilde{\boldsymbol{b}}_{t-1})' A_t^+ \boldsymbol{x}_t| \\
&\overset{(6.1)}{=} \sum_{t=1}^{T} \left| y_t \sum_{s=1}^{t-1} \gamma^{t-s} \boldsymbol{x}'_s A_t^+ \boldsymbol{x}_t \right| \overset{(5.1)}{\le} Y \sum_{t=1}^{T} \sum_{s=1}^{t-1} \gamma^{t-s} |\boldsymbol{x}'_s A_t^+ \boldsymbol{x}_t| \\
&\overset{\text{Lemma 6.1}}{\le} \frac{Y}{2} \sum_{t=1}^{T} \sum_{s=1}^{t-1} \gamma^{t-s} \boldsymbol{x}'_t A_t^+ \boldsymbol{x}_t + \frac{Y}{2} \sum_{t=1}^{T} \sum_{s=1}^{t-1} \gamma^{t-s} \boldsymbol{x}'_s A_s^+ \boldsymbol{x}_s \\
&= \frac{Y}{2} \sum_{t=1}^{T} \underbrace{\left(\sum_{s=1}^{t-1} \gamma^{t-s} \right)}_{\le \frac{\gamma}{1-\gamma}} \boldsymbol{x}'_t A_t^+ \boldsymbol{x}_t + \frac{Y}{2} \sum_{s=1}^{T} \underbrace{\left(\sum_{t=s+1}^{T} \gamma^{t-s} \right)}_{\le \frac{\gamma}{1-\gamma}} \boldsymbol{x}'_s A_s^+ \boldsymbol{x}_s \\
&\le Y \frac{\gamma}{1-\gamma} \sum_{t=1}^{T} \boldsymbol{x}'_t A_t^+ \boldsymbol{x}_t.
\end{aligned}$$

□

We can now show our main result.

Theorem 6.1. *Consider temporal-difference learning with $\gamma \in [0, 1)$ and $a \ge 0$. Let $(\boldsymbol{x}_1, r_1), (\boldsymbol{x}_2, r_2), \ldots$ be any sequence of examples in $\mathbb{R}^n \times \mathbb{R}$ such that the outcomes $y_1, y_2, y_3, \ldots$ given by (1.1) lie in the real interval $[-Y, Y]$. Then with the predictions $\hat{y}_t = C_Y(\tilde{\boldsymbol{b}}'_{t-1} A_t^+ \boldsymbol{x}_t)$ of the TLS algorithm,*

$$\begin{aligned}
&\sum_{t=1}^{T} (y_t - \hat{y}_t)^2 - \inf_{\boldsymbol{w} \in \mathbb{R}^n} \left(\sum_{t=1}^{T} (y_t - \boldsymbol{w} \cdot \boldsymbol{x}_t)^2 + a \|\boldsymbol{w}\|^2 \right) \\
&\quad \le Y^2 \left(1 + \frac{4\gamma}{1-\gamma} \right) \sum_{t=1}^{T} \boldsymbol{x}'_t A_t^+ \boldsymbol{x}_t.
\end{aligned}$$

Proof: Let $p_t := \boldsymbol{b}'_{t-1} A_t^+ \boldsymbol{x}_t$, $\tilde{p}_t := \tilde{\boldsymbol{b}}'_{t-1} A_t^+ \boldsymbol{x}_t$. Because of $y_t \in [-Y, Y]$ we know that $(y_t - C_Y(p_t))^2 \le (y_t - p_t)^2$, i.e. the relative loss bound of Theorem 3.2(i) also holds for

the clipped predictions $C_Y(p_t)$. Thus it suffices to show that

$$\sum_{t=1}^{T}(y_t - C_Y(\tilde{p}_t))^2 - \sum_{t=1}^{T}(y_t - C_Y(p_t))^2 \leq Y^2 \frac{4\gamma}{1-\gamma} \sum_{t=1}^{T} \boldsymbol{x}_t' A_t^+ \boldsymbol{x}_t.$$

This holds because

$$\begin{aligned}
&\sum_{t=1}^{T}((y_t - C_Y(\tilde{p}_t))^2 - (y_t - C_Y(p_t))^2) \\
&\quad = \sum_{t=1}^{T}(C_Y(\tilde{p}_t) - C_Y(p_t))(C_Y(\tilde{p}_t) + C_Y(p_t) - 2y_t) \\
&\quad \overset{(5.1)}{\leq} 4Y \sum_{t=1}^{T} |C_Y(\tilde{p}_t) - C_Y(p_t)| \leq 4Y \sum_{t=1}^{T} |\tilde{p}_t - p_t| \\
&\quad = 4Y \sum_{t=1}^{T} |\tilde{\boldsymbol{b}}_{t-1}' A_t^+ \boldsymbol{x}_t - \boldsymbol{b}_{t-1}' A_t^+ \boldsymbol{x}_t| \\
&\quad \overset{\text{Lemma 6.2}}{\leq} 4Y^2 \frac{\gamma}{1-\gamma} \sum_{t=1}^{T} \boldsymbol{x}_t' A_t^+ \boldsymbol{x}_t.
\end{aligned}$$

□

Note that the bound decays as the discount rate parameter γ approaches 1. In our definition of the outcome as a discounted geometric sum (1.1) we required that γ was less than one.

In the next two subsections we apply Theorem 6.1 to show relative loss bounds for the worst case and for the average case.

6.1. Worst case relative loss bound

The following corollary is, to our knowledge, the first logarithmic relative loss bound for the temporal-difference learning model we consider in this paper. Previously such strong relative loss bounds were only known for linear regression.

Corollary 6.1. *Consider temporal-difference learning with $\gamma \in [0, 1)$ and $a > 0$. Let $(\boldsymbol{x}_1, r_1), (\boldsymbol{x}_2, r_2), \ldots$ be any sequence of examples in $\mathbb{R}^n \times \mathbb{R}$ such that the outcomes $y_1, y_2, y_3, \ldots$ given by (1.1) lie in the real interval $[-Y, Y]$. Then with the predictions $\hat{y}_t = C_Y(\tilde{\boldsymbol{b}}_{t-1}' A_t^{-1} \boldsymbol{x}_t)$ of the TLS algorithm,*

$$\begin{aligned}
&\sum_{t=1}^{T}(y_t - \hat{y}_t)^2 - \inf_{\boldsymbol{w} \in \mathbb{R}^n} \left(\sum_{t=1}^{T}(y_t - \boldsymbol{w} \cdot \boldsymbol{x}_t)^2 + a\|\boldsymbol{w}\|^2 \right) \\
&\quad \leq Y^2 \left(1 + \frac{4\gamma}{1-\gamma}\right) \ln \det \left(\frac{1}{a} A_T\right) \\
&\quad \leq nY^2 \left(1 + \frac{4\gamma}{1-\gamma}\right) \ln \left(1 + \frac{TX^2}{an}\right),
\end{aligned}$$

where $X = \max_{t \in \{1,\ldots,T\}} \|\boldsymbol{x}_t\|$.

Proof: The first inequality follows from Theorem 6.1 and Theorem 3.2(ii). The second follows from Theorem 3.1. □

6.2. Average case relative loss bound

A typical assumption for convergence proofs in temporal-difference learning is that the sequence of instances $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ is a Markov chain. In this case we can apply the worst case relative loss bounds from Section 6.1.

Under the stronger assumption that the outcomes $y_1, y_2, \ldots$ lie in $[-Y, Y]$ and that the instances $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ are i.i.d. with some unknown distribution on $\mathbb{R}^n$, we can show an upper bound on the expectation of the relative loss (1.2) for trials 1 through T that only depends on n, Y, γ, and T. In particular we do not need the term $a\|\boldsymbol{w}\|^2$ that measures the complexity of the vector $\boldsymbol{w}$ in the relative loss (1.3), and we do not need to assume that the instances are bounded. Perhaps the most interesting consequence of this result is that the Bayes techniques that are usually used to show lower bounds on the relative loss for linear regression (see, e.g., Vovk (1997) or Section 9 of this paper) cannot be modified to give significantly stronger lower bounds.

To show the average case result of this section we will use Theorem 6.1 and will then bound sums of terms $\boldsymbol{x}_s' A_t^+ \boldsymbol{x}_s$, $s \in \{1, \ldots, t\}$, with the following theorem ($\mathrm{Tr}(A)$ is the trace of a square matrix A and $\dim(X)$ is the dimensionality of a vector space X).

Theorem 6.2. *For any t vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t \in \mathbb{R}^n$ with linear span $X_t = \mathrm{span}\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}$:*

$$\sum_{s=1}^{t} \boldsymbol{x}_s' A_t^+ \boldsymbol{x}_s = \dim(X_t) \le n, \qquad \text{if } a = 0,$$

$$\sum_{s=1}^{t} \boldsymbol{x}_s' A_t^+ \boldsymbol{x}_s = n - a\,\mathrm{Tr}\left(A_t^{-1}\right) \le \dim(X_t) \le n, \quad \text{if } a > 0.$$

Proof: We first look at the case $a = 0$. Let $m := \dim(X_t)$ and choose any orthonormal basis $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_m$ of X_t. Then

$$\forall \boldsymbol{x} \in X_t\colon\ \boldsymbol{x} = \sum_{1 \le i \le m} (\boldsymbol{x} \cdot \boldsymbol{e}_i)\boldsymbol{e}_i. \tag{6.2}$$

The above can also be written as

$$\forall \boldsymbol{x} \in X_t\colon\ \boldsymbol{x} = \left(\sum_{1 \le i \le m} \boldsymbol{e}_i \boldsymbol{e}_i'\right)\boldsymbol{x}. \tag{6.3}$$

This means that if we interpret the matrix $\sum_{1 \le i \le m} \boldsymbol{e}_i \boldsymbol{e}_i' \in \mathbb{R}^{n \times n}$ as a linear function from $\mathbb{R}^n$ to $\mathbb{R}^n$, it is the identity function on X_t. The assertion for the case $a = 0$ now follows

from

$$\begin{aligned}
\sum_{s=1}^{t} \boldsymbol{x}_s' A_t^+ \boldsymbol{x}_s &\overset{(6.2)}{=} \sum_{s=1}^{t} \sum_{1\le i,j\le m} (\boldsymbol{e}_i \cdot \boldsymbol{x}_s)\boldsymbol{e}_i' A_t^+ \boldsymbol{e}_j (\boldsymbol{x}_s \cdot \boldsymbol{e}_j) \\
&= \sum_{1\le i,j\le m} \boldsymbol{e}_i' A_t^+ \boldsymbol{e}_j \sum_{s=1}^{t} \boldsymbol{e}_i' \boldsymbol{x}_s \boldsymbol{x}_s' \boldsymbol{e}_j \overset{(2.4),a=0}{=} \sum_{1\le i,j\le m} \boldsymbol{e}_j' A_t^+ \boldsymbol{e}_i \boldsymbol{e}_i' A_t \boldsymbol{e}_j \\
&= \sum_{1\le j\le m} \boldsymbol{e}_j' A_t^+ \left(\sum_{1\le i\le m} \boldsymbol{e}_i \boldsymbol{e}_i' \right) A_t \boldsymbol{e}_j \overset{(6.3)}{=} \sum_{1\le j\le m} \boldsymbol{e}_j' A_t^+ A_t \boldsymbol{e}_j \\
&= \sum_{1\le j\le m} \boldsymbol{e}_j' \boldsymbol{e}_j = \sum_{1\le j\le m} 1 = m = \dim(X_t).
\end{aligned}$$

To prove the theorem for the case $a > 0$ we choose an arbitrary orthonormal basis $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n$ of $\mathbb{R}^n$ and apply the result for $a = 0$ to the vectors $\boldsymbol{x}_{1-n} := \sqrt{a}\boldsymbol{e}_1, \ldots, \boldsymbol{x}_0 := \sqrt{a}\boldsymbol{e}_n, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t$. First note that $\sum_{s=1-n}^{t} \boldsymbol{x}_s \boldsymbol{x}_s' = aI + \sum_{s=1}^{t} \boldsymbol{x}_s \boldsymbol{x}_s' = A_t$. From the case $a = 0$ we have $\sum_{s=1-n}^{t} \boldsymbol{x}_s' A_t^+ \boldsymbol{x}_s = n$ since the vectors $\{\boldsymbol{x}_{1-n}, \ldots, \boldsymbol{x}_t\}$ have rank n. The equality for $a > 0$ now follows from $\sum_{s=1-n}^{0} \boldsymbol{x}_s' A_t^+ \boldsymbol{x}_s = a \sum_{i=1}^{n} \boldsymbol{e}_i' A_t^{-1} \boldsymbol{e}_i = a \operatorname{Tr}(A_t^{-1})$. The first inequality for the case $a > 0$ follows from $a \operatorname{Tr}(A_t^{-1}) \ge \dim(X_t^\perp) = n - \dim(X_t)$, where we use Corollary A.1. □

Corollary 6.2. *Consider the temporal-difference learning setting with $\gamma \in [0, 1)$ and $a = 0$. Assume that the instances $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ are i.i.d. with unknown distribution on $\mathbb{R}^n$ and that the outcomes $y_1, y_2, \ldots$ given by* (1.1) *lie in the real interval $[-Y, Y]$. Then with the predictions $\hat{y}_t = C_Y(\tilde{\boldsymbol{b}}_{t-1}' A_t^+ \boldsymbol{x}_t)$ of the TLS algorithm, the expectation of* (1.2) *is*

$$\begin{aligned}
&\mathrm{E}\left(\sum_{t=1}^{T} (y_t - \hat{y}_t)^2 - \inf_{\boldsymbol{w}\in\mathbb{R}^n} \left(\sum_{t=1}^{T} (y_t - \boldsymbol{w}\cdot\boldsymbol{x}_t)^2 \right) \right) \\
&\quad \le Y^2 \left(1 + \frac{4\gamma}{1-\gamma} \right) \sum_{t=1}^{T} \frac{\mathrm{E}(\dim(\operatorname{span}\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}))}{t} \\
&\quad \le nY^2 \left(1 + \frac{4\gamma}{1-\gamma} \right) (1 + \ln T).
\end{aligned}$$

Proof: Because of Theorem 6.1 the expected relative loss is at most

$$Y^2 \left(1 + \frac{4\gamma}{1-\gamma} \right) \sum_{t=1}^{T} \mathrm{E}(\boldsymbol{x}_t' A_t^+ \boldsymbol{x}_t).$$

The instances $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t$ are i.i.d. and the matrix A_t^+ is a symmetric function of the $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t$. This implies that the expectation $\mathrm{E}(\boldsymbol{x}_t' A_t^+ \boldsymbol{x}_t)$ does not change if we swap $\boldsymbol{x}_t$ with

some $\boldsymbol{x}_s$, $s = 1, \ldots, t$. Thus $\mathrm{E}(\boldsymbol{x}_t' A_t^+ \boldsymbol{x}_t) = \mathrm{E}(\boldsymbol{x}_s' A_t^+ \boldsymbol{x}_s)$ for $s = 1, \ldots, t$, and

$$\begin{aligned}\mathrm{E}(\boldsymbol{x}_t' A_t^+ \boldsymbol{x}_t) &= \frac{1}{t} \sum_{s=1}^{t} \mathrm{E}(\boldsymbol{x}_s' A_t^+ \boldsymbol{x}_s) \\ &= \frac{1}{t} \mathrm{E}\left(\sum_{s=1}^{t} \boldsymbol{x}_s' A_t^+ \boldsymbol{x}_s \right) \leq \frac{\mathrm{E}(\dim(\mathrm{span}\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}))}{t},\end{aligned}$$

where we used Theorem 6.2 for the final inequality. This proves the first inequality of Corollary 6.2. The second follows from $\sum_{t=1}^{T} \frac{1}{t} \leq 1 + \ln T$. □

It should be straightforward to extend the result of Corollary 6.2 to the case of rapidly mixing Markov chains.

7. Episodic learning

Until now we studied a setting where the outcomes $y_t = \sum_{s=t}^{\infty} \gamma^{s-t} r_s$ are discounted sums of *all* future reinforcement signals. If we use our algorithm for policy evaluation in reinforcement learning, this corresponds to looking at *continuing tasks* (see Sutton & Barto, 1998). For our relative loss bounds we had to assume that the discount rate parameter γ is strictly smaller than 1. For the case $\gamma = 1$ we need to assume that the trials are partitioned into *episodes* of finite length, i.e. we consider *episodic tasks*. Now an outcome depends only on reinforcement signals that belong to the same episode. To show relative loss bounds we now assume that the lengths of the episodes are bounded.

Let t be a trial. The first trial that is in the same episode as trial t is denoted by $\mathrm{start}(t)$ and the last by $\mathrm{end}(t)$. With this notation and with a discount rate parameter $\gamma \in [0, 1]$, the outcome y_t in the episodic setting is defined as

$$y_t = \sum_{s=t}^{\mathrm{end}(t)} \gamma^{s-t} r_s. \tag{7.1}$$

This replaces the definition of y_t given in (1.1) for the continuous setting. The definitions of the relative loss (1.2) and (1.3) remain unchanged. Note that the continuous setting is essentially the episodic setting with one episode of infinite length.

With the same motivation as in Section 5 we get the TLS algorithm for episodic learning which is presented in Table 3. An implementation of this algorithm is given in Table 4. We can check the correctness of this implementation by verifying that

$$A_{\mathrm{inv}} = A_t^{-1}, \quad \boldsymbol{z} = \sum_{s=\mathrm{start}(t)}^{t} \gamma^{t-s} \boldsymbol{x}_s, \quad \tilde{\boldsymbol{b}} = \tilde{\boldsymbol{b}}_t$$

hold after every iteration of the FOR-loop. This follows from (2.5) and the equality

$$\tilde{\boldsymbol{b}}_t = \tilde{\boldsymbol{b}}_{t-1} + r_t \sum_{s=\mathrm{start}(t)}^{t} \gamma^{t-s} \boldsymbol{x}_s.$$

Table 3. The temporal least squares (TLS) algorithm for episodic learning.

At trial t, TLS predicts with

$$\hat{y}_t = C_Y(\tilde{\boldsymbol{b}}'_{t-1} A_t^{-1} \boldsymbol{x}_t),$$

where

$$A_t = aI + \sum_{s=1}^{t} \boldsymbol{x}_s \boldsymbol{x}'_s,$$

$$\tilde{\boldsymbol{b}}_t = \sum_{k=1}^{t} r_k \sum_{s=\text{start}(k)}^{k} \gamma^{k-s} \boldsymbol{x}_s.$$

C_Y given by (5.2) clips the prediction to the interval $[-Y, Y]$ and start(k) is the first trial in the same episode to which trial k belongs.

If A_t is not invertible, then the inverse A_t^{-1} of A_t must be replaced by the pseudoinverse A_t^{+}.

Table 4. Implementation of TLS for episodic learning.

$A_{\text{inv}} := \frac{1}{a} I \in \mathbb{R}^{n \times n}$

$\tilde{\boldsymbol{b}} := \boldsymbol{0} \in \mathbb{R}^n$

FOR $t = 1, 2, 3, \ldots$ DO

If a new episode starts at trial t, then set $\boldsymbol{z} := \boldsymbol{0} \in \mathbb{R}^n$

Receive instance vector $\boldsymbol{x}_t \in \mathbb{R}^n$

$$A_{\text{inv}} := A_{\text{inv}} - \frac{(A_{\text{inv}} \boldsymbol{x}_t)(A_{\text{inv}} \boldsymbol{x}_t)'}{1 + \boldsymbol{x}'_t A_{\text{inv}} \boldsymbol{x}_t}$$

Predict with $\hat{y}_t = C_Y(\tilde{\boldsymbol{b}}' A_{\text{inv}} \boldsymbol{x}_t) \in \mathbb{R}$

Receive reinforcement signal $r_t \in \mathbb{R}$

$\boldsymbol{z} := \gamma \boldsymbol{z} + \boldsymbol{x}_t$

$\tilde{\boldsymbol{b}} := \tilde{\boldsymbol{b}} + r_t \boldsymbol{z}$

END FOR

A note for the practitioner: If $a = 0$ and if t is small, then the matrix A_t might not be invertible and our algorithm uses the pseudoinverse of A_t. In practice we suggest to use $a > 0$ and tune this parameter. Then A_t is always invertible and the calculation of pseudoinverses can be avoided. We also conjecture that the clipping is not needed for practical data.

We now show relative loss bounds for episodic learning. Again we assume that a bound Y on the outcomes is known to the learner in advance. The proof of the following theorem is very similar to the proof of Theorem 6.1.

Theorem 7.1. *Consider temporal-difference learning with episodes of length at most ℓ and let $a \geq 0$. Let $(\boldsymbol{x}_1, r_1), (\boldsymbol{x}_2, r_2), \ldots$ be any sequence of examples in $\mathbb{R}^n \times \mathbb{R}$ such that the outcomes $y_1, y_2, y_3, \ldots$ given by* (7.1) *lie in the real interval $[-Y, Y]$. Then with the*

predictions $\hat{y}_t = C_Y(\tilde{\boldsymbol{b}}'_{t-1} A_t^+ \boldsymbol{x}_t)$ *of the TLS algorithm of Table* 3,

$$\sum_{t=1}^{T}(y_t - \hat{y}_t)^2 - \inf_{\boldsymbol{w}\in\mathbb{R}^n}\left(\sum_{t=1}^{T}(y_t - \boldsymbol{w}\cdot\boldsymbol{x}_t)^2 + a\|\boldsymbol{w}\|^2\right)$$
$$\leq Y^2\left(1 + 4\frac{\gamma-\gamma^l}{1-\gamma}\right)\sum_{t=1}^{T}\boldsymbol{x}'_t A_t^+ \boldsymbol{x}_t \tag{7.2}$$

If $a > 0$, *then* (7.2) *is bounded by*

$$nY^2\left(1 + 4\frac{\gamma-\gamma^l}{1-\gamma}\right)\ln\left(1 + \frac{TX^2}{an}\right),$$

where $X = \max_{t\in\{1,\ldots,T\}}\|\boldsymbol{x}_t\|$.

If the instances $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ *are i.i.d. with some unknown distribution on* $\mathbb{R}^n$, *then the expectation of* (7.2) *is at most*

$$nY^2\left(1 + 4\frac{\gamma-\gamma^l}{1-\gamma}\right)(1 + \ln T). \tag{7.3}$$

A lower bound corresponding to (7.3) is shown in Theorem 9.2.

Note that Theorem 7.1 does not exploit the fact that different episodes have varying length. Related theorems that depend on the actual lengths of the episodes can easily be developed.

8. Other second order algorithms

Second order algorithms for temporal-difference learning have been proposed by Bradtke and Barto (1996) and Boyan (1999). We compare the algorithms in the episodic setting. Their algorithms maintain weight vectors $\boldsymbol{w}_t$ and predict with $\hat{y}_t = \boldsymbol{w}_t \cdot \boldsymbol{x}_t$ at trial t.

Bradtke and Barto's Least-squares TD, or shorter LSTD, algorithm uses the weight vectors

$$\boldsymbol{w}_t = \left(\sum_{s=1}^{t-1}\boldsymbol{x}_s(\boldsymbol{x}_s - \gamma\boldsymbol{x}_{s+1})'\right)^{-1}\left(\sum_{s=1}^{t-1} r_s\boldsymbol{x}_s\right).$$

Boyan's LSTD(λ) algorithm (he only considers the case $\gamma = 1$) has a parameter $\lambda \in [0, 1]$ like the TD(λ) algorithm. It uses the weight vectors

$$\boldsymbol{w}_t = \left(\sum_{s=1}^{t-1}\tilde{\boldsymbol{z}}_s(\boldsymbol{x}_s - \boldsymbol{x}_{s+1})'\right)^{-1}\left(\sum_{k=1}^{t-1} r_k\tilde{\boldsymbol{z}}_k\right),$$

where

$$\tilde{\boldsymbol{z}}_k = \sum_{s=\mathrm{start}(k)}^{k}\lambda^{k-s}\boldsymbol{x}_s.$$

In contrast our TLS algorithm uses the weight vectors

$$\boldsymbol{w}_t = \left(aI + \sum_{s=1}^{t} \boldsymbol{x}_s \boldsymbol{x}_s' \right)^{-1} \left(\sum_{k=1}^{t-1} r_k \boldsymbol{z}_k \right)$$

where

$$\boldsymbol{z}_k = \sum_{s=\text{start}(k)}^{k} \gamma^{k-s} \boldsymbol{x}_s.$$

and the prediction of the TLS algorithm at trial t is $\hat{y}_t = C_Y(\boldsymbol{w}_t \cdot \boldsymbol{x}_t)$. (In the above formulas the inverse must be replaced by the pseudoinverse if the matrix is not invertible.)

Note that the TLS algorithm does not have a parameter λ like TD(λ) or LSTD(λ), and that for the case $\gamma = 1$ the algorithms LSTD and LSTD(0) are identical. An important difference of the TLS algorithm to TD(λ) and LSTD(λ) is that it does not use differences in the definition of the "covariance" matrix.

Bradtke, Barto and Boyan experimentally compare their algorithms to TD(λ). They find that their second order algorithms make much better use of the data than TD(λ). Under comparatively strong assumptions Bradtke and Barto also show that the $\boldsymbol{w}_t$ of their algorithm converge asymptotically.

The TLS algorithm we proposed in this paper was designed to minimize the relative loss (1.3), and our relative loss bounds show that TLS does this well. We do not know whether similar relative loss bounds hold for the LSTD and LSTD(λ) algorithms.

A small experimental comparison of Boyan's LSTD(λ) and our TLS algorithm on randomly generated acyclic Markov chains is given in Forster (2001). In the experiments Boyan's algorithm is unstable in the presence of noise, while our algorithm, even for $a = 0$, is much more stable. (This problem was already pointed out in Boyan's original paper.) However, if the matrix aI is incorporated into the prediction of Boyan's algorithm (as done in the prediction (5.5) of the TLS algorithm) then both algorithms show very similar performance. In that case Boyan's algorithm is slightly better on data with small noise, while the TLS algorithm performs better on very noisy data. Recall that the matrix aI in the prediction was caused by adding a regularization term $a\|\boldsymbol{w}\|^2$ to the motivation of the algorithm (see for example (1.3)).

9. Lower bounds

In this section we give lower bounds for linear regression and for episodic temporal-difference learning. First consider the case of linear regression, i.e. $\gamma = 0$. In this case the outcomes y_t are equal to the reinforcement signals. If the outcomes y_t lie in $[-Y, Y]$, then Corollary 6.2 gives the upper bound of

$$nY^2(1 + \ln T) \tag{9.1}$$

on the expected relative loss (1.2). Here the examples are i.i.d. with respect to an arbitrary distribution.

In the next theorem we show that the bound (9.1) cannot be improved substantially. Our proof is very similar to the proof of Theorem 2 of Vovk (1997). However, if the dimension n of the instances is greater than one, the examples in his proof are generated by a stochastic strategy and are not i.i.d.

Theorem 9.1. *Consider linear regression ($\gamma = 0$) with $a = 0$. For every $\varepsilon > 0$ there is a constant C and a distribution $\mathcal{D}$ on the set of examples $\mathbb{R}^n \times [-Y, Y]$ such that for all T and for every learning algorithm the expectation of the relative loss* (1.2) *is*

$$\mathrm{E}_{((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_T, y_T)) \sim \mathcal{D}^T} \left(\sum_{t=1}^{T} (y_t - \hat{y}_t)^2 - \inf_{\boldsymbol{w} \in \mathbb{R}^n} \left(\sum_{t=1}^{T} (y_t - \boldsymbol{w} \cdot \boldsymbol{x}_t)^2 \right) \right) \geq (n - \varepsilon) Y^2 \ln T - C,$$

where the examples $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_T, y_T)$ are i.i.d. with distribution $\mathcal{D}$.

Proof: For a fixed parameter $\alpha \geq 1$ we generate a distribution $\mathcal{D}$ on the examples with the following stochastic strategy: A vector $\boldsymbol{\theta} \in [0, 1]^n$ is chosen from the prior distribution $\mathrm{Beta}(\alpha, \alpha)^n$, i.e. the components of $\boldsymbol{\theta}$ are i.i.d. with distribution $\mathrm{Beta}(\alpha, \alpha)$. Then $\mathcal{D} = \mathcal{D}_{\boldsymbol{\theta}}$ is the distribution for which the example $(\boldsymbol{e}_i, 1)$ has probability $\frac{\theta_i}{n}$ and the example $(\boldsymbol{e}_i, -1)$ has probability $\frac{1-\theta_i}{n}$. Here $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n$ is the canonical base of $\mathbb{R}^n$. In each trial the examples are generated i.i.d. with $\mathcal{D}_{\boldsymbol{\theta}}$. We can calculate the Bayes optimal learning algorithm for which the expectation of the loss in trials 1 through T is minimal. The expectation of the relative loss of this algorithm gives the lower bound of Theorem 9.1.

Details of the proof are given in Appendix B. □

Now consider the episodic setting discussed in Section 7. Here the trials are partitioned into episodes and the outcomes are given by $y_t = \sum_{s=t}^{\mathrm{end}(t)} \gamma^{s-t} r_s$. The following lower bound is proven by a reduction to the previous lower bound for linear regression.

Theorem 9.2. *Consider episodic temporal-difference learning where all episodes have fixed length ℓ. Let $[-Y, Y]$ be a range for the outcomes. Then for every $\varepsilon > 0$ there is a constant C and a stochastic strategy that generates instances $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \in \mathbb{R}^n$ and reinforcement signals $r_1, r_2, \ldots$ such that the outcomes lie in $[-Y, Y]$ and for all T greater than ℓ and for every learning algorithm the expectation* (*over the stochastic choice of the examples*) *of the relative loss* (1.2) *is*

$$\mathrm{E}\left(\sum_{t=1}^{T} (y_t - \hat{y}_t)^2 - \inf_{\boldsymbol{w} \in \mathbb{R}^n} \left(\sum_{t=1}^{T} (y_t - \boldsymbol{w} \cdot \boldsymbol{x}_t)^2 \right) \right) \geq \frac{1 - \gamma^{2\ell}}{1 - \gamma^2} \left(Y^2 (n - \varepsilon) \ln \left\lfloor \frac{T}{\ell} \right\rfloor - C \right).$$

Proof: We modify the stochastic strategy used in the proof of Theorem 9.1. When this strategy generates an instance $\boldsymbol{x}_t$ and an outcome y_t in trial t, we now generate a whole episode of ℓ trials with instances $\gamma^{\ell-1}\boldsymbol{x}_t, \ldots, \gamma\boldsymbol{x}_t, \boldsymbol{x}_t$ and reinforcement signals $0, \ldots, 0, y_t$. The outcomes for this episode are $\gamma^{\ell-1}y_t, \ldots, \gamma y_t, y_t$.

Consider just the q-th trials from each episode for some $1 \leq q \leq \ell$. In these trials the learner essentially processes the scaled examples $\gamma^{\ell-q}(\boldsymbol{x}_t, y_t)$. So the lower bound of Theorem 9.1 applies with a factor of $\gamma^{2(\ell-q)}$. All ℓ choices of q lead to a factor of $1 + \gamma^2 + \cdots + \gamma^{2(\ell-1)}$ in the lower bound. □

10. Conclusion and open problems

We proposed a new algorithm for temporal-difference learning, the TLS algorithm. Contrary to previous second order algorithms, the new algorithm does not use differences in the definition of its "covariance matrix", see discussion in Section 8. The main question is whether these differences are really helpful. We proved worst case relative loss bounds for the TLS algorithm. It would be interesting to know how tight our bounds are for some practical data.

In this paper we focus on worst case relative loss bounds for temporal difference learning. However, we also use our techniques to prove tight relative loss bounds for the average case under an i.i.d. assumption. It should be straightforward to extend our results to the case of rapidly mixing Markov chains.

In our bounds the class of linear functions serves as a comparison class. We use a second order algorithm and its additional loss over the loss of the best comparator is logarithmic in the number of trials. We conjecture that even for linear regression there is no first order algorithm with adaptive learning rates for which the additional loss is logarithmic in the number of trials.

The algorithms analyzed here can be applied to the case when the instances are expanded to feature vectors and the dot product between two feature vectors is given by a kernel function (see Saunders, Gammerman, & Vovk, 1998). Also Fourier or wavelet transforms can be used to extract information from the instances, see Walker (1996) and Graps (1995). One can reduce the dimensionality of the comparison class by only using a subset of the coefficients.

So far we compared the total loss of the on-line algorithm to the total loss of the best linear predictor on the whole sequence of examples. Now suppose that the comparator is produced by partitioning the data sequence into k segments and picking the best linear predictor for each segment. Again we aim to bound the total loss of the on-line algorithm minus the total loss of the best comparator of this form. Such bound have been obtained by Herbster and Warmuth (1998) for the case of linear regression using first-order algorithms. We would like to know whether there is a simple second-order algorithm for linear regression that requires $O(n^2)$ update time per trial and for which the additional loss grows with the sums of the logs of the section lengths.

Most of our paper focused on continuous learning, where each outcome is an infinite discounted sum of future reinforcement signals. In Section 7 we discussed how the TLS algorithm can be adapted to the episodic setting. Here the outcomes only depend on

reinforcement signals from the same episode:

$$y_t = \sum_{s=t}^{\mathrm{end}(t)} \gamma^{s-t} r_s.$$

For some applications it might make more sense to let the outcomes y_t be convex combinations of the future reinforcement signals of the episode and define

$$y_t = \frac{\sum_{s=t}^{\mathrm{end}(t)} \gamma^{t-s} r_s}{\sum_{s=t}^{\mathrm{end}(t)} \gamma^{t-s}}. \tag{10.1}$$

In the case when $\gamma = 1$ each outcome would be the average of the future reinforcement signals. We do not know of any relative loss bounds for the case when y_t is defined as (10.1).

On a more technical level we would like to know if it is really necessary to clip the predictions of the temporal-difference algorithm we proposed. Our proofs are reductions to the previous proofs for linear regression. Direct proofs might avoid clipping.

Another open technical question is discussed at the end of Section 3. We conjecture that the parameter a in Vovk's linear regression algorithm can be tuned to obtain bounds of the form (3.3) proven for the (first order) Widrow-Hoff algorithm. Similarly we believe that the parameter a in the new (second order) learning algorithm of the paper can be tuned to obtain the bound (4.1) proven for the (first order) TD$^*(\lambda)$ algorithm of Schapire and Warmuth.

Finally, note that we do not have lower bounds for the continuous setting with $\gamma > 0$. It should be possible to show a lower bound of $(n-\varepsilon)\frac{1}{1-\gamma}Y^2 \ln T - C$ on the expected relative loss (1.2). (See Theorem 9.2 for a corresponding lower bound in the episodic case.)

Appendix A: Properties of pseudoinverses

In Section 2 we used the pseudoinverse of the matrix A_t to find a vector $w \in \mathbb{R}^n$ that minimizes (2.2) in the case $a = 0$. Here we look at the pseudoinverse of A_t in some detail.

We begin by showing that A_t (as defined in (2.4)) is always invertible on the subspace $X_t = \mathrm{span}\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}$ of $\mathbb{R}^n$.

Lemma A.1. *Let $a \in [0, \infty)$ and let $X_t^\perp$ be the orthogonal complement of X_t in $\mathbb{R}^n$. Then the linear map*

$$A_t : X_t \to X_t$$

is invertible and

$$A_t : X_t^\perp \to X_t^\perp$$

is equal to aI.

Proof: A_t maps vectors from X_t to X_t because for $\boldsymbol{x} \in X_t$

$$A_t \boldsymbol{x} = a\boldsymbol{x} + \sum_{s=1}^{t} (\boldsymbol{x} \cdot \boldsymbol{x}_s)\boldsymbol{x}_s \in X_t.$$

$A_t : X_t \to X_t$ is one-to-one because if $A_t \boldsymbol{x} = \boldsymbol{0}$ holds for a vector $\boldsymbol{x} \in X_t$ it follows that

$$0 = \boldsymbol{x}' A_t \boldsymbol{x} = a\|\boldsymbol{x}\|^2 + \sum_{s=1}^{t} \boldsymbol{x}' \boldsymbol{x}_s \boldsymbol{x}_s' \boldsymbol{x} = a\|\boldsymbol{x}\|^2 + \sum_{s=1}^{t} (\boldsymbol{x}_s \cdot \boldsymbol{x})^2,$$

i.e. $\boldsymbol{x} \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}^\perp = X_t^\perp$. Thus $\boldsymbol{x} \in X_t \cap X_t^\perp = \{\boldsymbol{0}\}$. Since $\dim X_t$ is finite this also implies that $A_t : X_t \to X_t$ is onto.

Finally we have that for $\boldsymbol{x} \in X_t^\perp$:

$$A_t \boldsymbol{x} = a\boldsymbol{x} + \sum_{s=1}^{t} \underbrace{(\boldsymbol{x} \cdot \boldsymbol{x}_s)}_{=0} \boldsymbol{x}_s = a\boldsymbol{x} = (aI)\boldsymbol{x}. \qquad \square$$

Now we can calculate the pseudoinverse A_t^+ of the linear map A_t:

Corollary A.1. *For all $a \in [0, \infty)$, the matrix A_t^+ is positive semidefinite and satisfies*

$$\forall \boldsymbol{x} \in X_t: \quad A_t^+ A_t \boldsymbol{x} = \boldsymbol{x} = A_t A_t^+ \boldsymbol{x},$$

$$\forall \boldsymbol{x} \in X_t^\perp: \; A_t^+ \boldsymbol{x} = \begin{cases} \frac{1}{a}\boldsymbol{x}, & a > 0, \\ \boldsymbol{0}, & a = 0. \end{cases}$$

Proof: Since A_t maps X_t into X_t and $X_t^\perp$ into $X_t^\perp$ (by Lemma A.1) we can calculate the pseudoinverse on X_t and on $X_t^\perp$ separately. Now we only have to note that the pseudoinverse of aI is $\frac{1}{a} I$ if $a > 0$ and it is the zero matrix if $a = 0$, and that the pseudoinverse of an invertible matrix is the inverse matrix. $\square$

We show some results for the case $a > 0$, and then argue that they also hold for $a = 0$. For this the following lemma is helpful.

Lemma A.2. *For $\boldsymbol{x} \in X_t$ and $\boldsymbol{z} \in \mathbb{R}^n$, the term $\boldsymbol{x}' A_t^+ \boldsymbol{z}$ is continuous in $a \in [0, \infty)$.*

Proof: Let $\mathcal{G}$ be the space of invertible linear functions from X_t to X_t. The function $[0, \infty) \to \mathbb{R}$, $a \mapsto \boldsymbol{x}' A_t^+ \boldsymbol{z}$ is the composition of the following four continuous functions:

- The function $[0, \infty) \to \mathcal{G}$, $a \mapsto A_T$ (see Lemma A.1).
- The function $\mathcal{G} \to \mathcal{G}$, $A \mapsto A^{-1}$ (for the continuity of this function see, e.g., Bollobás, 1999, Chapter 12, Corollary 3).
- The function $\mathcal{G} \to X_t$, $A \mapsto A\boldsymbol{x}$.
- The function $X_t \to \mathbb{R}$, $\boldsymbol{u} \mapsto \boldsymbol{u} \cdot \boldsymbol{z}$. $\square$

The following two technical lemmas are needed in the case when an instance $\boldsymbol{x}_t$ does not lie in the linear span of the previous instances $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}$.

Lemma A.3. *If $a = 0$ and $\boldsymbol{x}_t \notin X_{t-1}$, then $A_t^+ \boldsymbol{x}_t \in X_{t-1}^\perp$.*

Proof: Because of $X_t \backslash X_{t-1} \neq \emptyset$ there exists a $\boldsymbol{z} \in X_t \cap X_{t-1}^\perp$, $\boldsymbol{z} \neq \boldsymbol{0}$. For this $\boldsymbol{z}$

$$A_t \boldsymbol{z} \overset{z \in X_{t-1}^\perp}{=} \boldsymbol{x}_t \boldsymbol{x}_t' \boldsymbol{z} = (\boldsymbol{z} \cdot \boldsymbol{x}_t) \boldsymbol{x}_t.$$

Applying A_t^+ to the above equality $(\boldsymbol{z} \cdot \boldsymbol{x}_t)\boldsymbol{x}_t = A_t \boldsymbol{z}$ gives

$$(\boldsymbol{z} \cdot \boldsymbol{x}_t) A_t^+ \boldsymbol{x}_t = A_t^+ A_t \boldsymbol{z} \overset{\boldsymbol{z} \in X_t, \text{Corollary A.1}}{=} \boldsymbol{z}.$$

This equality together with $\boldsymbol{z} \neq \boldsymbol{0}$ implies that $\boldsymbol{z} \cdot \boldsymbol{x}_t \neq 0$. Thus

$$A_t^+ \boldsymbol{x}_t = (\boldsymbol{z} \cdot \boldsymbol{x}_t)^{-1} \boldsymbol{z} \overset{\boldsymbol{z} \in X_{t-1}^\perp}{\in} X_{t-1}^\perp. \qquad \square$$

We write $a \searrow 0$ to denote that $a \in (0, \infty)$ goes to zero.

Lemma A.4. *If $\boldsymbol{x}_t \notin X_{t-1}$ then $\boldsymbol{x}_t' A_{t-1}^{-1} \boldsymbol{x}_t \to \infty$ for $a \searrow 0$.*

Proof: Because of $X_{t-1} + X_{t-1}^\perp = \mathbb{R}^n$ we can write $\boldsymbol{x}_t = \boldsymbol{b} + \boldsymbol{c}$ with $\boldsymbol{b} \in X_{t-1}, \boldsymbol{c} \in X_{t-1}^\perp$. Because of $\boldsymbol{x}_t \notin X_{t-1}$ we know that $\boldsymbol{c} \neq \boldsymbol{0}$. By Corollary A.1, $\boldsymbol{x}_t' A_{t-1}^{-1} \boldsymbol{x}_t$ is equal to

$$\boldsymbol{b}' A_{t-1}^{-1} \boldsymbol{b} + \boldsymbol{c}' A_{t-1}^{-1} \boldsymbol{c} = \boldsymbol{b}' A_{t-1}^{-1} \boldsymbol{b} + \frac{1}{a} \|\boldsymbol{c}\|^2.$$

Because of $\boldsymbol{c} \neq \boldsymbol{0}$ this goes to infinity as $a \searrow 0$. $\square$

Appendix B: Proof of Theorem 9.1

Let $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n$ be the canonical base of $\mathbb{R}^n$. We show that for every $\varepsilon > 0$ there is a constant C' and a distribution $\mathcal{D}$ on $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\} \times \{0, 1\}$ such that

$$\mathrm{E}_{((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_T, y_T)) \sim \mathcal{D}^T} \left(\sum_{t=1}^T (y_t - \hat{y}_t)^2 - \inf_{\boldsymbol{w} \in \mathbb{R}^n} \left(\sum_{t=1}^T (y_t - \boldsymbol{w} \cdot \boldsymbol{x}_t)^2 \right) \right)$$
$$\geq \frac{1}{4}(n - \varepsilon) \ln T - C'. \tag{B.1}$$

In the above the outcomes y_t lie in $\{0, 1\}$. For the transformed outcomes $(2y_t - 1)Y \in [-Y, Y]$ the lower bound becomes $Y^2(n - \varepsilon) \ln T - 4Y^2 C'$.

We will use the beta distribution to generate the distribution $\mathcal{D}$. For parameters $\alpha, \beta > 0$ the distribution $\mathrm{Beta}(\alpha, \beta)$ has the density function

$$\frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha, \beta)}, \quad \theta \in [0, 1],$$

with regard to the Lebesgue measure on $[0, 1]$. Here

$$B(\alpha, \beta) = \int_0^1 \theta^{\alpha-1}(1-\theta)^{\beta-1} \, \mathrm{d}\theta = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha + \beta)} \tag{B.2}$$

is the beta function, and Γ is the gamma function which satisfies

$$\Gamma(\alpha) = (\alpha - 1)\Gamma(\alpha - 1), \quad \alpha \in \mathbb{R} \setminus \{1, 0, -1, -2, -3, \ldots\}.$$

It can easily be seen that

$$\mathrm{E}_{\theta\sim\mathrm{Beta}(\alpha,\alpha)}(\theta-\theta^2)=\frac{\alpha}{4\alpha+2}. \tag{B.3}$$

We generate examples with the following stochastic strategy: Fix a parameter $\alpha > 0$ that we choose later. First a vector $\boldsymbol{\theta} \in [0,1]^n$ is chosen from the distribution $\mathrm{Beta}(\alpha,\alpha)^n$, i.e. the components of $\boldsymbol{\theta}$ are i.i.d. with distribution $\mathrm{Beta}(\alpha,\alpha)$. Then the examples are generated i.i.d. with the distribution $\mathcal{D}_{\boldsymbol{\theta}}$ on $\{\boldsymbol{e}_1,\ldots,\boldsymbol{e}_n\}\times\{0,1\}$ for which

$$\mathcal{D}_{\boldsymbol{\theta}}\{(\boldsymbol{e}_i,0)\}=\frac{1-\theta_i}{n},\quad \mathcal{D}_{\boldsymbol{\theta}}\{(\boldsymbol{e}_i,1)\}=\frac{\theta_i}{n}$$

for $i = 1,\ldots,n$. For this stochastic strategy we can calculate the Bayes optimal algorithm, i.e. the learning algorithm for which the expectation (over the prior distribution of $\boldsymbol{\theta}$) of the expected relative loss (B.1) (with $\mathcal{D}=\mathcal{D}_{\boldsymbol{\theta}}$) is minimal.

We use the notation

$$K_t:=\sum_{s=1}^{t-1}\boldsymbol{x}_s\cdot\boldsymbol{x}_t,\quad k_t:=\sum_{s=1}^{t-1}y_s\boldsymbol{x}_s\cdot\boldsymbol{x}_t.$$

K_t counts how often $\boldsymbol{x}_t$ occurs among the instances $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{t-1}$, and k_t counts how often the example $(\boldsymbol{x}_t,1)$ occurs among the examples $(\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_{t-1},y_{t-1})$.

The expectation of the loss of a learning algorithm in trials 1 through T is

$$\begin{aligned}
&\mathrm{E}_{\boldsymbol{\theta}\sim\mathrm{Beta}(\alpha,\alpha)^n}\mathrm{E}_{((\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_T,y_T))\sim\mathcal{D}_{\boldsymbol{\theta}}^T}\left(\sum_{t=1}^{T}(\hat{y}_t-y_t)^2\right)\\
&\quad=\sum_{t=1}^{T}\mathrm{E}_{\boldsymbol{\theta}\sim\mathrm{Beta}(\alpha,\alpha)^n}\mathrm{E}_{((\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_t,y_t))\sim\mathcal{D}_{\boldsymbol{\theta}}^t}((\hat{y}_t-y_t)^2)\\
&\quad=\sum_{t=1}^{T}\int_{[0,1]^n}\prod_{i=1}^{n}\frac{\left(\theta_i-\theta_i^2\right)^{\alpha-1}}{B(\alpha,\alpha)}\sum_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_t\in\{\boldsymbol{e}_1,\ldots,\boldsymbol{e}_n\}}\sum_{y_1,\ldots,y_t\in\{0,1\}}\\
&\qquad\times\prod_{i=1}^{n}\left(\frac{1-\theta_i}{n}\right)^{\sum_{s=1}^{t}(1-y_s)\boldsymbol{x}_s\cdot\boldsymbol{e}_i}\left(\frac{\theta_i}{n}\right)^{\sum_{s=1}^{t}y_s\boldsymbol{x}_s\cdot\boldsymbol{e}_i}(\hat{y}_t-y_t)^2\,\mathrm{d}\boldsymbol{\theta}\\
&\quad=\sum_{t=1}^{T}\frac{1}{n^t}\sum_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_t\in\{\boldsymbol{e}_1,\ldots,\boldsymbol{e}_n\}}\sum_{y_1,\ldots,y_t\in\{0,1\}}\\
&\qquad\times\left(\prod_{i=1}^{n}\frac{B(\alpha+\sum_{s=1}^{t}(1-y_s)\boldsymbol{x}_s\cdot\boldsymbol{e}_i,\alpha+\sum_{s=1}^{t}y_s\boldsymbol{x}_s\cdot\boldsymbol{e}_i)}{B(\alpha,\alpha)}\right)(\hat{y}_t-y_t)^2.
\end{aligned}$$

From this formula we see what the best learning algorithm for this setting is: For fixed $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_t$ and $y_1,\ldots,y_{t-1}$ the sum of the terms that depend on the prediction $\hat{y}_t(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_t,y_1,\ldots,y_{t-1})$ is a positive constant times

$$B(\alpha+K_t-k_t+1,\alpha+k_t)\hat{y}_t^2+B(\alpha+K_t-k_t,\alpha+k_t+1)(\hat{y}_t-1)^2.$$

This is minimal if

$$\hat{y}_t(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, y_1, \ldots, y_{t-1}) = \left(\frac{B(\alpha + K_t - k_t + 1, \alpha + k_t)}{B(\alpha + K_t - k_t, \alpha + k_t + 1)} + 1\right)^{-1} = \left(\frac{\alpha + K_t - k_t}{\alpha + k_t} + 1\right)^{-1} = \frac{k_t + \alpha}{K_t + 2\alpha}. \tag{B.4}$$

Now we calculate the expected loss of the optimal algorithm. Let $\mathrm{Ber}(p)$ denote the Bernoulli distribution on $\{0, 1\}$ with mean $p \in [0, 1]$. The expected loss of the learning algorithm given in (B.4) is

$$\begin{aligned}
&\mathrm{E}_{\boldsymbol{\theta}\sim\mathrm{Beta}(\alpha,\alpha)^n}\mathrm{E}_{((\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_T,y_T))\sim\mathcal{D}_\theta^T}\left(\sum_{t=1}^{T}\left(\frac{k_t+\alpha}{K_t+2\alpha}-y_t\right)^2\right)\\
&\quad=\sum_{t=1}^{T}\frac{1}{n^t}\sum_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_t\in\{\boldsymbol{e}_1,\ldots,\boldsymbol{e}_n\}}\mathrm{E}_{\boldsymbol{\theta}\sim\mathrm{Beta}(\alpha,\alpha)^n}\\
&\qquad\times\mathrm{E}_{y_1\sim\mathrm{Ber}(\boldsymbol{\theta}\cdot\boldsymbol{x}_1)}\cdots\mathrm{E}_{y_t\sim\mathrm{Ber}(\boldsymbol{\theta}\cdot\boldsymbol{x}_t)}\left(\left(\frac{k_t+\alpha}{K_t+2\alpha}-y_t\right)^2\right)\\
&\quad=\frac{\alpha}{4\alpha+2}\sum_{t=1}^{T}\frac{1}{n^t}\sum_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_t\in\{\boldsymbol{e}_1,\ldots,\boldsymbol{e}_n\}}\left(1+\frac{1}{K_t+2\alpha}\right).
\end{aligned}$$

To see that the last equality holds, fix $\boldsymbol{\theta}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t$. The instance $\boldsymbol{x}_t$ is some unit vector $\boldsymbol{e}_i$. Thus $\boldsymbol{\theta}\cdot\boldsymbol{x}_t = \theta_i$. Since

$$\begin{aligned}
&\mathrm{E}_{y_t\sim\mathrm{Ber}(\boldsymbol{\theta}\cdot\boldsymbol{x}_t)}(y_t) = \mathrm{E}_{y_t\sim\mathrm{Ber}(\boldsymbol{\theta}\cdot\boldsymbol{x}_t)}\big(y_t^2\big) = \boldsymbol{\theta}\cdot\boldsymbol{x}_t = \theta_i,\\
&\mathrm{E}_{y_1\sim\mathrm{Ber}(\boldsymbol{\theta}\cdot\boldsymbol{x}_1)}\cdots\mathrm{E}_{y_{t-1}\sim\mathrm{Ber}(\boldsymbol{\theta}\cdot\boldsymbol{x}_{t-1})}(k_t) = K_t\theta_i,\\
&\mathrm{E}_{y_1\sim\mathrm{Ber}(\boldsymbol{\theta}\cdot\boldsymbol{x}_1)}\cdots\mathrm{E}_{y_{t-1}\sim\mathrm{Ber}(\boldsymbol{\theta}\cdot\boldsymbol{x}_{t-1})}\big(k_t^2\big) = \big(K_t^2 - K_t\big)\theta_i^2 + K_t\theta_i,
\end{aligned}$$

it follows that

$$\begin{aligned}
&\mathrm{E}_{y_1\sim\mathrm{Ber}(\boldsymbol{\theta}\cdot\boldsymbol{x}_1)}\cdots\mathrm{E}_{y_T\sim\mathrm{Ber}(\boldsymbol{\theta}\cdot\boldsymbol{x}_t)}\left(\left(\frac{k_t+\alpha}{K_t+2\alpha}-y_t\right)^2\right)\\
&\quad=\frac{\big(K_t^2-K_t\big)\theta_i^2+K_t\theta_i+2\alpha K_t\theta_i+\alpha^2}{(K_t+2\alpha)^2}-2\theta_i\frac{K_t\theta_i+\alpha}{K_t+2\alpha}+\theta_i\\
&\quad=\theta_i-\theta_i^2+\frac{1}{(K_t+2\alpha)^2}\big[\theta_i^2\big((K_t+2\alpha)^2+K_t^2-K_t-2K_t(K_t+2\alpha)\big)\\
&\qquad+\theta_i(K_t+2\alpha K_t-2\alpha(K_t+2\alpha))+\alpha^2\big]\\
&\quad=\big(\theta_i-\theta_i^2\big)\left(1+\frac{K_t-4\alpha^2}{(K_t+2\alpha)^2}\right)+\frac{\alpha^2}{(K_t+2\alpha)^2}
\end{aligned}$$

Integrating the above term over $\boldsymbol{\theta} \in \text{Beta}(\alpha, \alpha)^n$ gives

$$\frac{\alpha}{4\alpha+2}\left(1 + \frac{K_t - 4\alpha^2 + 4\alpha^2 + 2\alpha}{(K_t + 2\alpha)^2}\right) = \frac{\alpha}{4\alpha+2}\left(1 + \frac{1}{K_t + 2\alpha}\right),$$

where we used (B.3).

The expected value of the comparison term for fixed $\boldsymbol{\theta}$ is at most

$$\begin{aligned}
&\mathrm{E}_{((\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_T,y_T))\sim\mathcal{D}_\theta^T}\left(\inf_{\boldsymbol{w}\in\mathbb{R}^n}\sum_{t=1}^T(\boldsymbol{w}\cdot\boldsymbol{x}_t - y_t)^2\right)\\
&\quad\le \inf_{\boldsymbol{w}\in\mathbb{R}^n}\mathrm{E}_{((\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_T,y_T))\sim\mathcal{D}_\theta^T}\left(\sum_{t=1}^T(\boldsymbol{w}\cdot\boldsymbol{x}_t - y_t)^2\right)\\
&\quad= T\inf_{\boldsymbol{w}\in\mathbb{R}^n}\mathrm{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_\theta}((\boldsymbol{w}\cdot\boldsymbol{x} - y)^2)\\
&\quad= T\inf_{\boldsymbol{w}\in\mathbb{R}^n}\sum_{i=1}^n\left(\frac{1-\theta_i}{n}\boldsymbol{w}_i^2 + \frac{\theta_i}{n}(\boldsymbol{w}_i - 1)^2\right)\\
&\quad= T\inf_{\boldsymbol{w}\in\mathbb{R}^n}\frac{1}{n}\sum_{i=1}^n\left(\boldsymbol{w}_i^2 - \theta_i\boldsymbol{w}_i^2 + \theta_i\boldsymbol{w}_i^2 - 2\theta_i\boldsymbol{w}_i + \theta_i\right)\\
&\quad= \frac{T}{n}\inf_{\boldsymbol{w}\in\mathbb{R}^n}\sum_{i=1}^n((\boldsymbol{w}_i - \theta_i)^2 + \theta_i(1-\theta_i))\\
&\quad= \frac{T}{n}\sum_{i=1}^n\theta_i(1-\theta_i)
\end{aligned}$$

Integrating this over $\boldsymbol{\theta} \sim \text{Beta}(\alpha, \alpha)^n$ and using Equality (B.3) gives $\frac{\alpha}{4\alpha+2}T$.

We have shown that for any learning algorithm the expected relative loss is at least

$$\begin{aligned}
&\mathrm{E}_{\boldsymbol{\theta}\sim\text{Beta}(\alpha,\alpha)^n}\mathrm{E}_{((\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_T,y_T))\sim\mathcal{D}_\theta^T}\left(\sum_{t=1}^T(\hat{y}_t - y_t)^2 - \inf_{\boldsymbol{w}\in\mathbb{R}^n}\sum_{t=1}^T(\boldsymbol{w}\cdot\boldsymbol{x}_t - y_t)^2\right)\\
&\quad\ge \frac{\alpha}{4\alpha+2}\sum_{t=1}^T\frac{1}{n^t}\sum_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_t\in\{\boldsymbol{e}_1,\ldots,\boldsymbol{e}_n\}}\left(\frac{1}{K_t + 2\alpha}\right)\\
&\quad= \frac{\alpha}{4\alpha+2}\sum_{t=1}^T\mathrm{E}\left(\frac{1}{S_{t-1} + 2\alpha}\right),
\end{aligned} \tag{B.5}$$

where S_t is the sum of t independent random variables with distribution $\text{Ber}(\frac{1}{n})$. $\text{Ber}(\frac{1}{n})$ has mean $\frac{1}{n}$ and variance $\frac{1}{n} - \frac{1}{n^2} = \frac{n-1}{n^2}$. Thus $\mathrm{E}(\frac{S_{t-1}}{t-1}) = \frac{1}{n}$, $\mathrm{V}(\frac{S_{t-1}}{t-1}) = \frac{n-1}{(t-1)n^2}$. Chebychev's inequality

$$P(|X - \mathrm{E}(X)| \ge b) \le \frac{\mathrm{V}(X)}{b^2}$$

applied to the random variable $X = \frac{S_{t-1}}{t-1}$ and to $b = \frac{\delta}{n}$ shows that

$$P\left(\left|\frac{S_{t-1}}{t-1} - \frac{1}{n}\right| \geq \frac{\delta}{n}\right) \leq \frac{n-1}{(t-1)\delta^2}$$

for any $\delta > 0$. This means that $S_{t-1} < \frac{t-1}{n}(1+\delta)$ holds with probability at least $1 - \frac{n-1}{(t-1)\delta^2}$. That is, for a given $\varepsilon > 0$, (B.5) is at least

$$\begin{aligned}
&\frac{\alpha}{4\alpha+2}\frac{n}{1+\delta}\sum_{t=1}^{T}\left(1 - \frac{n-1}{(t-1)\delta^2}\right)\frac{1}{t-1+\frac{2\alpha n}{1+\delta}} \\
&\quad\geq \frac{1}{4}(n-\varepsilon)\sum_{t=T_0}^{T}\frac{1}{t-1+\frac{2\alpha n}{1+\delta}} \\
&\quad\geq \frac{1}{4}(n-\varepsilon)\ln T - C'.
\end{aligned}$$

Here we choose α large, δ small, and T_0 large for the first inequality, and C' large for the last inequality. □

Acknowledgments

Jürgen Forster was supported by a "DAAD Doktorandenstipendium im Rahmen des gemeinsamen Hochschulsonderprogramms III von Bund und Ländern", by the Deutsche Forschungsgemeinschaft, grant SI 498/4–1, and by a grant from the G.I.F., the German-Israeli Foundation for Scientific Research and Development. Manfred Warmuth was supported by the NSF grant CCR-9821087. Thanks to Nigel Duffy and the anonymous referees for valuable comments.

Note

1. Alternatively y_t could be defined as $(1-\gamma)\sum_{s=t}^{\infty}\gamma^{s-t}r_s$, which makes y_t a convex combination of the reinforcement signals $r_t, r_{t+1}, \ldots$ The alternate definition amounts to a simple rescaling of the outcomes and predictions.

References

Azoury, K., & Warmuth, M. K. (2001). Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning, 43:3*, 211–246.

Bollobás, B. (1999). *Linear analysis: An introductory course*. Cambridge: Cambridge University Press.

Boyan, J. (1999). Least-squares temporal difference learning. In *Proceedings of the Sixteenth International Conference on Machine Learning* (pp. 63–70). San Francisco: Morgan Kaufmann.

Bradtke, S. J., & Barto, A. G. (1996). Linear least-squares algorithms for temporal difference learning. *Machine Learning, 22:1/3*, 33–57.

Cesa-Bianchi, N., Long, P., & Warmuth, M. K. (1996). Worst-case quadratic loss bounds for on-line predictions of linear functions by gradient descent. *IEEE Transactions on Neural Networks, 7*, 604–619.

Forster, J. (1999). On relative loss bounds in generalized linear regression. In *Proceedings of the Twelfth International Symposium on Fundamentals of Computation Theory* (pp. 269–280). Berlin: Springer.

Forster, J. (2001). Ph.D. Thesis, Fakultät für Mathematik, Ruhr-Universität Bochum.

Foster, D. P. (1991). Prediction in the worst case. *The Annals of Statistics, 19*, 1084–1090.

Graps, A. L. (1995). An introduction to wavelets. *IEEE Computational Sciences and Engineering, 2*, 50–61.

Hassibi, B., Kivinen, J., & Warmuth, M. K. (1995). Unpublished manuscript. Department of Computer Science, University of California, Santa Cruz.

Herbster, M., & Warmuth, M. K. (1998). Tracking the best regressor. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory* (pp. 24–31). New York: ACM Press.

Kivinen, J., & Warmuth, M. K. (1997). Additive versus exponentiated gradient updates for linear prediction. *Journal of Information and Computation, 132*, 1–64.

Press, W. H., Flannery, B. P., Teukolsky, S. A., & Vetterling, W. T. (1989). *Numerical recipes in pascal*. Cambridge: Cambridge University Press.

Rektorys, K. (1994). *Survey of applicable mathematics*, 2nd rev. edn. Boston: Kluwer Academic Publishers.

Saunders, C., Gammerman, A., & Vovk, V. (1998). Ridge regression learning algorithm in dual variables. In *Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 515–521). San Francisco: Morgan Kaufmann.

Schapire, R. E., & Warmuth, M. K. (1996). On the worst-case analysis of temporal-difference learning algorithms. *Machine Learning, 22:1/3*, 95–121.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning, 3:1*, 9–44.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

Vovk, V. (1997). Competitive on-line linear regression. Technical Report CSD-TR-97-13, Department of Computer Science, Royal Holloway, University of London.

Walker, J. S. (1996). *Fast Fourier transforms*, 2nd edn. New York: CRC Press.

Widrow, B., & Stearns, S. (1985). *Adaptive signal processing*. Englewood Cliffs, NJ: Prentice-Hall.

Received July 17, 2000
Revised January 23, 2002
Accepted January 23, 2002
Final manuscript January 23, 2002