A new optimal step size for approximate dynamic programming

Ilya O. Ryzhov  Peter I. Frazier  Warren B. Powell

February 28, 2012

Abstract

Approximate dynamic programming (ADP) has proven itself in a wide range of applications spanning large scale transportation problems, scheduling, health care and energy systems. The design of effective ADP algorithms has many dimensions, but one crucial factor is the stepsize rule used to update a value function approximation. Many operations research applications are computationally intensive, and it is important to obtain good results quickly. Furthermore, the most popular stepsize formulas use tunable parameters and can produce very poor results if tuned improperly. We derive a new convergent stepsize rule that optimizes the prediction error in order to improve the short-term performance of an ADP algorithm. Without any tunable parameters, the new rule adapts to the level of noise in the measurements and produces faster convergence in a wide range of problems.

1 Introduction

Approximate dynamic programming (ADP) has emerged as a powerful tool for solving stochastic optimization problems in machine scheduling (Ronconi & Powell, 2010), inventory control (Adelman & Klabjan, 2012), health care (He et al., 2010; Maxwell et al., 2010), energy storage (Lai et al., 2010; Löhndorf & Minner, 2010) and sensor management (Castanon, 1997). In recent research, ADP has been used to solve a large-scale fleet management problem with 50,000 variables per time period and millions of dimensions in the state variable (Simão et al., 2009), and an energy resource planning problem with 175,000 time periods (Powell et al., 2011). Applications in operations research are especially demanding, often requiring the sequential solution of linear, nonlinear or integer programming problems. When an ADP algorithm is limited to a few hundred iterations, it is important to find a good solution as quickly as possible, a process that hinges on a stepsize (or learning rate) which controls how new information is merged with existing estimates.

We illustrate the learning process using the language of classical Markov decision processes. Consider an infinite-horizon dynamic program where $V(S_t)$ is the value of being in state $S_t \in \mathcal{S}$ and $C(S_t, x_t, W_{t+1})$ is a possibly random reward earned from being in state $S_t$, taking action $x_t \in \mathcal{X}$.


and then observing random information $W_{t+1}$. It is well-known (Howard, 1971; Puterman, 1994) that we can find the optimal, infinite-horizon value of being in each state using value iteration, which requires iteratively computing for each state $S_t \in \mathcal{S}$

$$V^n(S_t) = \max_{x_t \in \mathcal{X}} \mathbb{E} \left[ C(S_t, x_t, W_{t+1}) + \gamma V^{n-1}(S_{t+1}(S_t, x_t, W_{t+1})) \right] | S_t$$

(1)

where $\gamma < 1$ is a discount factor and $S_{t+1}$ is a random variable describing the state at time $t + 1$ given that we were in state $S_t$, took action $x_t$, and observed $W_{t+1}$ with values in some set $\mathcal{W}$.

There are many problems where (1) is difficult to solve due to the curse of dimensionality. For this reason, there is a tradition, dating back to Bellman’s earliest work (Bellman & Dreyfus, 1959), of solving this equation approximately. This field has evolved under a variety of names including approximate dynamic programming (ADP), neuro-dynamic programming and reinforcement learning (see Bertsekas & Tsitsiklis, 1996; Sutton & Barto, 1998; Si et al., 2004; Powell, 2007). In one major class of ADP methods known as approximate value iteration, an observation of the value $V(S_t)$ is bootstrapped from an approximation of the downstream value of $S_{t+1}$, and then used to update that approximation. A generic procedure for computing the observation is given by

$$\hat{v}_t^n = \max_{x_t} \sum_{w \in \mathcal{W}} P(W_{t+1} = w | S_t^n, x_t) \left[ C(S_t^n, x_t, w) + \gamma \bar{V}_{t+1}^{n-1}(S_{t+1}^n(S_t^n, x_t, w)) \right],$$

(2)

where $\bar{V}^{n-1}$ is the value function approximation, and $S_t^n \in \mathcal{S}$ is our state at time $t$ during the $n$th iteration of the ADP algorithm. We intend (2) only to illustrate the concept of constructing $\hat{v}_t^n$ from $\bar{V}_{t+1}^{n-1}$; in practice, the summation in (2) is difficult to compute (e.g. if $W_{t+1}$ follows a continuous distribution) and also needs to be approximated. The expectation can often be avoided entirely using the post-decision state concept introduced by Van Roy et al. (1997).

Regardless of the particular technique used, we update $\bar{V}^{n-1}$ by smoothing it with the new observation $\hat{v}_t^n$, obtaining

$$\bar{V}_t^n(S_t^n) = (1 - \alpha_{n-1}) \bar{V}_t^{n-1}(S_t^n) + \alpha_{n-1} \hat{v}_t^n$$

(3)

where $0 < \alpha_{n-1} \leq 1$ is a stepsize (or learning rate). Note again that (3) uses statistical bootstrapping, where the estimate of the value $\hat{v}_t^n$ depends on a statistical approximation $\bar{V}^{n-1}(S_t)$. This is the defining characteristic of approximate value iteration, which has proven to be very successful in broad classes of operations research applications. The reinforcement learning community uses a closely related algorithm known as Q-learning (Watkins & Dayan, 1992), which uses a similar bootstrapping scheme to learn the value of a state-action pair. In both approximate value iteration
and Q-learning, the stepsize plays two roles. First, it smooths out the effects of noise in our observations (the lower the stepsize, the smoother the approximation). Second, it determines how much weight is placed on the one-period rewards (the higher the stepsize, the more a reward is worth). This dual role of the stepsize is specific to bootstrapping-based methods, whose ease of use makes them a natural approach for large-scale OR applications where speed is crucial. See e.g. Topaloglu & Powell (2006) or ch. 12 of Powell (2007) for more examples of such applications.

The method of using a stepsize to update the value of being in state $S_t$ and taking action $x_t$ is based on the field of stochastic approximation; see Wasan (1969) and Kushner & Yin (1997) for thorough treatments of this field. Tsitsiklis (1994) and Jaakkola et al. (1994) were the first to apply this theory to prove the convergence of an ADP algorithm when the stepsize rule satisfies certain conditions. One example of such a stepsize rule is $\alpha_{n-1} = 1/n$, which has the effect of averaging over the observations $\hat{v}^n$. If the observations $\hat{v}^n$ are stationary (e.g. $\gamma = 0$), then $1/n$ is optimal in the sense of producing an estimate $\bar{V}^n$ with the lowest possible variance (see Kmenta, 1997). Since the $1/n$ rule satisfies the necessary theoretical conditions, it has become a kind of default rule used by beginners and theoreticians (see e.g. Azar et al., 2011, for a recent example). However, the $1/n$ rule can produce very slow convergence. The seminal work by Even-Dar & Mansour (2003) provides general bounds on the amount of time required to approach an optimal solution with a certain probability under the Q-learning algorithm. Other results on slow convergence can be found in Szepesvári (1997), Beleznay et al. (1999), Borkar (2002), and Gosavi (2008). One of our contributions in this paper is to introduce new bounds that are easier to compute and provide insights into the weakness of the $1/n$ stepsize.

In general, ADP practice shows a strong bias toward simple rules that are easy to code. One very popular stepsize is a simple constant such as $\alpha_{n-1} \equiv 0.1$. However, it is easy to construct problems where any single constant will work poorly. Powell & Simão (2009) solves an inventory problem for spare parts, where a high-volume spare part may remain in inventory for just a few days, while a low-volume part may remain in inventory for hundreds of days. A small stepsize will work very poorly with a low-volume part, while large stepsizes fail to dampen the noise, and would not be appropriate for high-volume parts. Such problems require the use of stochastic stepsize rules, where $\alpha_{n-1}$ is computed adaptively from the error in the previous prediction or estimate. These methods include the stochastic gradient rule of Benveniste et al. (1990) and the Delta-Bar-Delta rule of Sutton (1992), as well as the Kalman filter (Stengel, 1994; Choi & Van Roy, 2006).
detailed survey of both deterministic and stochastic stepsizes is given in George & Powell (2006). The latter work adapts the Kalman filter idea to create the Optimal Stepsize Algorithm (OSA). The OSA rule (also known as the bias-adjusted Kalman filter or BAKF rule) is optimal in the sense that, at time \( n \), it chooses the stepsize that minimizes the expected squared error of the \( n \)th prediction. However, the model used by OSA/BAKF comes from the signal processing literature, where the observations may be non-stationary, but are assumed to be mutually independent. This is not the case in dynamic programming, where the observation \( \hat{v}_n \) depends on the previous value function approximation, which itself depends on the previous observation.

We approach the problem of stepsize selection by studying an MDP with a single state and action, a problem that is designed to mimic the steady-state behaviour of a general MDP, thus distilling a large class of problems to an easily analyzed archetype. We make the following contributions: 1) Using the single state-action model, we derive easily computable, convergent upper and lower bounds on the time required for convergence under \( 1/n \), demonstrating that the rate of convergence of \( 1/n \) can be so slow that this rule should almost never be used for approximate value iteration. 2) Motivated by the slow convergence of \( 1/n \), we derive a stepsize rule that is optimal for our baseline approximate value iteration problem. Our analysis is similar to the derivation of BAKF in George & Powell (2006), but takes into consideration the specific updating structure of (2) and (3), which causes the observations to be dependent. To our knowledge, this is the first stepsize rule that strikes a balance between smoothing out noise and adding contributions over a horizon. 3) We analyze the convergence properties of our stepsize rule. We show that it does not stall, and declines to zero in the limit. This is the first optimal stepsize that provably has these properties, as BAKF was not shown to converge to zero. 4) We show how our stepsize rule can be easily adapted to a general ADP setting and demonstrate that, while popular competing strategies are sensitive to tunable parameters, our new rule is robust and does not require tuning. This last property is of vital practical importance, allowing developers to focus on testing approximation strategies without the concern that poor performance may be due to a poorly tuned stepsize formula.

In Section 2, we give our analysis of the slow convergence of approximate value iteration under the stepsize \( \alpha_{n-1} = 1/n \). Section 3 derives the optimal stepsize rule for the approximate value iteration problem. Section 4 discusses the experimental performance of the new rule in the approximate value iteration problem. Finally, Sections 5-6 show how the stepsize can be used in a more general ADP setting, and presents experimental comparisons with other stepsizes.
2 Motivation: slow convergence of $\alpha_{n-1} = 1/n$

The $1/n$ stepsize rule, while widely recognized by experimentalists to be slow, seems to persist in the literature as a kind of default rule, as evidenced by its use in the recent “speedy Q-learning algorithm” (Azar et al., 2011). In this section, we derive new analytical results on slow convergence of $1/n$ for algorithms that use bootstrapping.

Our analysis is based on a simple instance of the approximate value iteration problem with a single state and a single action. In this setting, (1) reduces to $v^* = c + \gamma v^*$ and has the solution $v^* = \frac{c}{1 - \gamma}$. The ADP equations (2) and (3) reduce to

$$\hat{v}^n = \hat{c}^n + \gamma \bar{v}^{n-1}, \quad (4)$$
$$\bar{v}^n = (1 - \alpha_{n-1}) \bar{v}^{n-1} + \alpha_{n-1} \hat{v}^n, \quad (5)$$

where the random variables $\hat{c}^n$, $n = 1, 2, \ldots$ are independent and identically distributed. We let $c = \mathbb{E}(\hat{c}^n)$ and $\sigma^2 = \text{Var}(\hat{c}^n)$.

There are important reasons to study this seemingly trivial MDP. First, recall the well-known property of Markov decision processes that a) the policy produced by the basic value iteration update in (1) converges to an optimal policy and b) the probability that we are in some state $s$ quickly converges to a steady state distribution (see Puterman, 1994). As a result, the expected contribution earned at each iteration approaches a constant that we can denote by $c$. That is, an infinite-horizon MDP resembles the single-state, single-action problem in the limit.

Second, the single-state, single-action MDP allows us to obtain important insights into more general ADP problems. In this section, we derive easily computable performance bounds for approximate value iteration under $\alpha_{n-1} = \frac{1}{n}$. These bounds imply that the $1/n$ stepsize rule can converge very slowly even in the single-state, single-action setting, instantly raising warning flags for attempting to use this rule in more complicated MDPs. The single-state, single-action case also allows us to show that $1/n$ is more pervasive in ADP than one might think. For a more in-depth discussion of this issue, see Appendix 9, where we demonstrate the role of $1/n$ in the popular method of recursive least-squares.

For the remainder of this section, we will assume that $\hat{c}^n = c$ for all $n$, that is, the rewards are deterministic. If an algorithm performs badly in this deterministic case, we generally expect
it to perform even worse when $\hat{c}$ is allowed to be random, since increasing noise generally slows convergence. We briefly summarize our results and give a numerical illustration. The reader is referred to Appendix 8 for the technical details. Our analysis is based on extending $\bar{v}^n$ from the natural numbers onto the positive reals through a piecewise linear interpolation. We define

$$\bar{v}(n) = (1 - (n - \lfloor n \rfloor))\bar{v}[\lfloor n \rfloor] + (n - \lfloor n \rfloor)\bar{v}[\lceil n \rceil]$$  \hspace{1cm} (6)$$

$$= \bar{v}[\lfloor n \rfloor] + (n - \lfloor n \rfloor)(\bar{v}[\lceil n \rceil] - \bar{v}[\lfloor n \rfloor])$$  \hspace{1cm} (7)$$

for all $n \in \mathbb{R}_+$, where $\bar{v}(n)$ is given for $n \in \mathbb{N}^*$ by the recursion defined by (5). Here, $\lfloor n \rfloor$ is the greatest integer less than or equal to $n$, and $\lceil n \rceil$ is the least integer greater than or equal to $n$. As can be seen in (6), we are simply writing $\bar{v}(n)$ as a weighted average of its rounded values.

First, we note some well-known properties of the sequence $\{\bar{v}(n)\}$ in Lemma 1, after which we state the lower and upper bounds. The proof of Lemma 1 is straightforward, and we omit it. The proofs of other results in this section are given in Appendix 8.

**Lemma 1.** $\bar{v}$ is increasing and concave in $n$, and bounded above by $\frac{c}{\gamma}$. 

**Theorem 1.** $\bar{v}(n) \geq \frac{c}{1-\gamma} (1 - (n+1)^{(1-\gamma)})$ for all $n \geq 0$.

**Lemma 2.** For $n \in \mathbb{N}^*$, $\frac{d}{dn} \bar{v}(n) \leq \frac{\bar{v}(n+1)}{n+1}$. 

**Theorem 2.** For any $n_0 > 0$, $\bar{v}$ is bounded above by

$$\bar{v}(n) \leq \frac{c}{1-\gamma} \left[ 1 - bn^{-(1-\gamma)} - \frac{1-\gamma}{\gamma} \frac{1}{n} \right]$$

where

$$b = n_0^{1-\gamma} \left[ 1 - \frac{1-\gamma}{n_0\gamma} - \frac{1-\gamma}{c} \bar{v}(n_0) \right].$$  \hspace{1cm} (8)$$

To make this upper bound concrete, we must choose $n_0$. The choice $n_0 = 1$ provides a practical upper bound.

**Corollary 1.** $\bar{v}(n) \leq \frac{c}{1-\gamma} \left[ 1 - bn^{-(1-\gamma)} - \frac{1-\gamma}{\gamma} \frac{1}{n} \right]$ for all $n \geq 1$ where $b = \frac{\gamma^2 + \gamma - 1}{\gamma}$.

Finally, we compare the upper and lower bounds against $\bar{v}$ computed numerically. We fix $c$ to 1, because it only enters as a multiplicative factor in the bounds and in the true value function as well. Thus $\gamma$ is our only free parameter. The results are plotted on a log-scale in Figure 1. As $n$
Figure 1: $\bar{v}(n)$ and its upper and lower bounds; computed for $\gamma = .7, .8, .9, .95$.

Figure 2: Upper and lower bounds on number of iterations needed to get within 1% of optimal, plotted for different ranges of $\gamma$.

grows large the upper and lower bounds come together and approach the limiting $v^* = 1/(1 - \gamma)$ value. Convergence slows as $\gamma$ increases.
In Figure 2, we calculate the number of iterations before $\bar{v}(n)$ reaches 1% of optimal. The lower bound on the value of $\bar{v}(n)$ gives an upper bound on the number of iterations needed, and the upper bound on $\bar{v}(n)$ gives a lower bound on the iterations needed. For $\gamma$ near .7, we already require 10,000 iterations, causing difficulty for applications requiring a significant amount of time per iteration. Then, as $\gamma$ grows larger than .8 we require at least 100 million iterations, which is impractical for almost any applications. As $\gamma$ grows to .9 and beyond the number of iterations needed is at least $10^{19}$.

We see that, in this simple problem, approximate value iteration with stepsize $1/n$ converges so slowly as to be impractical for most infinite horizon applications, particularly when the discount factor is close to 1. This behaviour is likely to be seen in other more complex infinite horizon problems, and also in undiscounted finite horizon problems. The infinite horizon problem we studied, which earns $c$ in every time period with discount factor $\gamma$, has the same reward, $c/(1 - \gamma)$, as an undiscounted finite horizon problem earning the same $c$ per time period with horizon length $T = 1/(1 - \gamma)$. Inverting $T = 1/(1 - \gamma)$ defines an “effective discount factor” $(T - 1)/T$ for finite horizon problems. As $T$ grows large, the effective discount factor grows close to 1, and approximate value iteration with a $1/n$ stepsize becomes intractably slow. The remainder of this paper studies a new stepsize rule that is optimal for the single-state, single-action MDP used in the above analysis.

3 An optimal stepsize for approximate value iteration

To explain our approach, let us first review the BAKF rule of George & Powell (2006). In the original derivation, this rule was known as the Optimal Stepsize Algorithm. Because it is not optimal for the particular problem we will consider, we will refer to it by the alternate name of “bias-adjusted Kalman filter” given in Powell (2007). The BAKF rule is designed for a signal processing problem, in which there is a sequence of independent observations $\hat{X}^n$ with unknown means $\theta^n$ and common variance $\sigma^2$. The unknown means are estimated by the usual exponential smoothing technique

$$\tilde{\theta}^n (\alpha_{n-1}) = (1 - \alpha_{n-1}) \tilde{\theta}^{n-1} + \alpha_{n-1} \hat{X}^n.$$ 

To compute $\tilde{\theta}^n$, the $n$th approximation, the BAKF rule chooses $\alpha_{n-1}$ to minimize

$$\min_{0 \leq \alpha_{n-1} \leq 1} \mathbf{E} \left[ (\tilde{\theta}^n (\alpha_{n-1}) - \theta^n)^2 \right].$$
The solution to this problem is given explicitly by the formula

\[
\alpha_{n-1} = 1 - \frac{\sigma^2}{(1 + \zeta^{n-1}) \sigma^2 + (\beta^n)^2}
\]  

(9)

where \(\zeta^{n-1}\) is given by the recursive formula

\[
\zeta^n = \begin{cases} 
\alpha^2_0 & n = 1 \\
\alpha^2_{n-1} + (1 - \alpha_{n-1})^2 \zeta^{n-1} & n > 1
\end{cases}
\]

and \(\beta^n = \theta^n - \mathbb{E} \hat{\theta}^{n-1}\) is the bias in the smoothed estimate from the previous iteration.

The BAKF rule is particularly relevant to our study because it chooses the stepsize to minimize the expected squared error of each prediction. However, BAKF has two limitations when applied specifically to dynamic programming. First, BAKF assumes (as is common in signal processing) that the observations used in smoothing are independent. This is clearly not the case in ADP. Even in the simple MDP considered in Section 2, we can easily see from (4) and (5) that \(\bar{v}^n\) depends on \(\bar{v}^{n-1}\), which in turn depends on \(\hat{v}^{n-1}\). Secondly, the bias \(\beta^n\) used to compute the BAKF stepsize is unknown in a general dynamic programming problem, making it necessary to estimate it using a second stochastic approximation (which requires a second stepsize rule). The BAKF rule is particularly sensitive to our ability to estimate \(\beta^n\), which limits the effectiveness of BAKF for high-noise applications.

In this section, we derive a new stepsize rule that is optimal specifically for the approximate value iteration problem given by (4) and (5). Our analysis offers two important advantages over BAKF. First, it includes an extra term that accounts for the covariance between \(\hat{v}^n\) and \(\bar{v}^{n-1}\). This term does not appear in the BAKF formula. To our knowledge, this is the first stepsize rule that takes the covariance structure of dynamic programming into consideration. Secondly, the simplicity of the single-state, single-action MDP allows us to compute the bias \(\beta^n\) exactly. This computation can then be used as an approximation for the bias in more general dynamic programming problems, thus eliminating the need to estimate the bias adaptively using a second stepsize.

3.1 Derivation

The approximate value iteration problem is given by (4) and (5). As before, let \(c = \mathbb{E} \hat{c}^n\) and \(\sigma^2 = Var(\hat{c}^n)\). Observe that \(\hat{v}^n\) can be written recursively as

\[
\hat{v}^n = (1 - \alpha_{n-1}) \hat{v}^{n-1} + \alpha_{n-1} \bar{c}^n + \alpha_{n-1} \gamma \bar{v}^{n-1} = (1 - (1 - \gamma) \alpha_{n-1}) \hat{v}^{n-1} + \alpha_{n-1} \bar{c}^n.
\]  

(10)
The particular structure of this problem allows us to derive recursive formulas for the mean and variance of the approximation \( \bar{v}^n \). We assume that \( \bar{v}^0 = 0 \).

**Proposition 1.** Define

\[
\delta^n = \begin{cases} 
\alpha_0 & n = 1 \\
\alpha_{n-1} + (1 - (1 - \gamma) \alpha_{n-1}) \delta^{n-1} & n > 1
\end{cases}
\]

and

\[
\lambda^n = \begin{cases} 
\alpha_0^2 & n = 1 \\
\alpha_{n-1}^2 + (1 - (1 - \gamma) \alpha_{n-1})^2 \lambda^{n-1} & n > 1
\end{cases}
\]

Then, \( \mathbb{E} (\bar{v}^n) = \delta^n c \) and \( \text{Var} (\bar{v}^n) = \lambda^n \sigma^2 \).

**Proof:** Observe that \( \mathbb{E} (\bar{v}^1) = \alpha_1 c = \delta^1 c \) and \( \text{Var} (\bar{v}^1) = \alpha_0^2 \sigma^2 = \lambda^1 c \). Now suppose that \( \mathbb{E} (\bar{v}^{n-1}) = \delta^{n-1} c \) and \( \text{Var} (\bar{v}^{n-1}) = \lambda^{n-1} \sigma^2 \). By (10), we have

\[
\mathbb{E} (\bar{v}^n) = (1 - (1 - \gamma) \alpha_{n-1}) \delta^{n-1} c + \alpha_{n-1} c = \delta^n c.
\]

Furthermore, \( \bar{v}^{n-1} \) depends only on \( \bar{v}^{n'} \) for \( n' < n \), therefore \( \bar{v}^{n-1} \) and \( \bar{v}^n \) are independent. Consequently,

\[
\text{Var} (\bar{v}^n) = (1 - (1 - \gamma) \alpha_{n-1})^2 \lambda^{n-1} \sigma^2 + \alpha_{n-1}^2 \sigma^2 = \lambda^n \sigma^2
\]

as required. \( \square \)

From Proposition 1, we can also compute explicit formulas

\[
\delta^n = \sum_{i=0}^{n-1} \alpha_i \prod_{j=i+1}^{n-1} (1 - (1 - \gamma) \alpha_j)
\]

and

\[
\lambda^n = \sum_{i=0}^{n-1} \alpha_i^2 \prod_{j=i+1}^{n-1} (1 - (1 - \gamma) \alpha_j)^2
\]

for \( \delta^n \) and \( \lambda^n \) in terms of the stepsizes. The next result shows that these quantities are uniformly bounded in \( n \).

**Proposition 2.** For all \( n \), \( \delta^n \leq \frac{1}{1 - \gamma} \) and \( \lambda^n \leq \frac{1}{\gamma (1 - \gamma)} \).
**Proof:** This is clearly true for \( n = 1 \), since \( \alpha_0 \leq 1 \). Suppose now that \( \delta^{n-1} \leq \frac{1}{\gamma(1-\gamma)} \) and \( \lambda^{n-1} \leq \frac{1}{\gamma(1-\gamma)} \) for \( n > 1 \). Then, using the definition of \( \delta^n \), we obtain

\[
\delta^n \leq \alpha_{n-1} + (1 - (1 - \gamma) \alpha_{n-1}) \frac{1}{1 - \gamma} = \frac{1}{1 - \gamma}.
\]

Similarly, we can write

\[
\lambda^n \leq \alpha_{n-1}^2 + (1 - (1 - \gamma) \alpha_{n-1})^2 \frac{1}{\gamma(1-\gamma)}
\]

\[
= \frac{1}{\gamma(1-\gamma)} + \left(1 + \frac{1-\gamma}{\gamma} \right) \alpha_{n-1}^2 - \frac{2}{\gamma} \alpha_{n-1}
\]

\[
= \frac{1}{\gamma(1-\gamma)} + \frac{1}{\gamma} \alpha_{n-1}^2 - \frac{2}{\gamma} \alpha_{n-1}
\]

\[
\leq \frac{1}{\gamma(1-\gamma)},
\]

as required. \( \square \)

We define the optimal stepsize for time \( n \) to be the value that achieves

\[
\min_{\alpha_{n-1} \in [0,1]} \mathbb{E} \left[ (\bar{v}^n(\alpha_{n-1}) - \mathbb{E}\bar{v}^n)^2 \right],
\]

which is the minimum squared deviation of the time-\( n \) estimate \( \bar{v}^n \) from the mean of the time-\( n \) observation \( \hat{v}^n \). We can simplify this objective function in the following manner:

\[
\mathbb{E} \left[ (\bar{v}^n(\alpha_{n-1}) - \mathbb{E}\bar{v}^n)^2 \right] = \mathbb{E} \left[ (1 - \alpha_{n-1}) \bar{v}^{n-1} + \alpha_{n-1} \hat{v}^n - \mathbb{E}\bar{v}^n)^2 \right]
\]

\[
= \mathbb{E} \left[ (1 - \alpha_{n-1}) (\hat{v}^{n-1} - \mathbb{E}\hat{v}^n) + \alpha_{n-1} (\hat{v}^n - \mathbb{E}\hat{v}^n))^2 \right]
\]

\[
= (1 - \alpha_{n-1})^2 \mathbb{E} \left[ (\hat{v}^{n-1} - \mathbb{E}\hat{v}^n)^2 \right] + \alpha_{n-1}^2 \mathbb{E} \left[ (\hat{v}^n - \mathbb{E}\hat{v}^n)^2 \right]
\]

\[
+ 2\alpha_{n-1} (1 - \alpha_{n-1}) \mathbb{E} \left[ (\hat{v}^{n-1} - \mathbb{E}\hat{v}^n)(\hat{v}^n - \mathbb{E}\hat{v}^n) \right].
\]

The first equality is obtained using the recursive formula for \( \bar{v}^n \) from (10). Observe that

\[
\mathbb{E} \left[ (\hat{v}^{n-1} - \mathbb{E}\hat{v}^n)(\hat{v}^n - \mathbb{E}\hat{v}^n) \right] = \mathbb{E} (\hat{v}^{n-1}\hat{v}^n) - \mathbb{E}\hat{v}^{n-1}\mathbb{E}\hat{v}^n
\]

\[
= \text{Cov}(\hat{v}^{n-1}, \hat{v}^n),
\]

whence we obtain

\[
\mathbb{E} \left[ (\bar{v}^n(\alpha_{n-1}) - \mathbb{E}\bar{v}^n)^2 \right] = (1 - \alpha_{n-1})^2 \mathbb{E} \left[ (\hat{v}^{n-1} - \mathbb{E}\hat{v}^n)^2 \right] + \alpha_{n-1}^2 \mathbb{E} \left[ (\hat{v}^n - \mathbb{E}\hat{v}^n)^2 \right]
\]

\[
+ 2\alpha_{n-1} (1 - \alpha_{n-1}) \text{Cov}(\hat{v}^{n-1}, \hat{v}^n).
\]
Proposition 3. The objective function in (11) is convex in $\alpha_{n-1}$.

Proof: Let $f(\alpha_{n-1})$ be the right-hand side of (12). First, observe that

$$\frac{d^2f}{d\alpha_{n-1}^2} = 2\mathbb{E}[(\hat{v}^{n-1} - \mathbb{E}\hat{v}^n)^2] + 2\mathbb{E}[(\hat{v}^n - \mathbb{E}\hat{v}^n)^2] - 4\text{Cov}(\hat{v}^{n-1}, \hat{v}^n).$$

It is enough to show that

$$2\text{Cov}(\hat{v}^{n-1}, \hat{v}^n) \leq \mathbb{E}[(\hat{v}^{n-1} - \mathbb{E}\hat{v}^n)^2] + \mathbb{E}[(\hat{v}^n - \mathbb{E}\hat{v}^n)^2].$$

We compute

$$\text{Cov}(\hat{v}^{n-1}, \hat{v}^n) = \mathbb{E}(\hat{v}^{n-1}\hat{v}^n) - \mathbb{E}\hat{v}^{n-1}\mathbb{E}\hat{v}^n$$

$$= \mathbb{E}(\hat{v}^{n-1}(\hat{v}^n + \gamma\hat{v}^{n-1})) - \mathbb{E}\hat{v}^{n-1}\mathbb{E}(\hat{v}^n + \gamma\hat{v}^{n-1})$$

$$= c\mathbb{E}\hat{v}^{n-1} + \gamma\mathbb{E}(\hat{v}^{n-1})^2 - c\mathbb{E}\hat{v}^{n-1} - \gamma(\mathbb{E}\hat{v}^{n-1})^2$$

$$= \gamma\text{Var}(\hat{v}^{n-1}), \quad (13)$$

where we use the independence of $\hat{v}^{n-1}$ and $\hat{v}^n$ to obtain the third line. We now use a bias-variance decomposition (see e.g. Hastie et al., 2001) to write

$$\mathbb{E}[(\hat{v}^{n-1} - \mathbb{E}\hat{v}^n)^2] = \mathbb{E}[(\hat{v}^{n-1} - \mathbb{E}\hat{v}^{n-1} + \mathbb{E}\hat{v}^{n-1} - \mathbb{E}\hat{v}^n)^2]$$

$$= \mathbb{E}[(\hat{v}^{n-1} - \mathbb{E}\hat{v}^{n-1})^2] + (\mathbb{E}\hat{v}^{n-1} - \mathbb{E}\hat{v}^n)^2$$

$$= \text{Var}(\hat{v}^{n-1}) + (\mathbb{E}\hat{v}^{n-1} - \mathbb{E}\hat{v}^n)^2 \quad (14)$$

where the cross term vanishes because the quantity $\mathbb{E}\hat{v}^{n-1} - \mathbb{E}\hat{v}^n$ is deterministic, and thus

$$\mathbb{E}[(\hat{v}^{n-1} - \mathbb{E}\hat{v}^{n-1})(\mathbb{E}\hat{v}^{n-1} - \mathbb{E}\hat{v}^n)] = (\mathbb{E}\hat{v}^{n-1} - \mathbb{E}\hat{v}^n)\mathbb{E}(\hat{v}^{n-1} - \mathbb{E}\hat{v}^{n-1}) = 0.$$

Observe that

$$\mathbb{E}[(\hat{v}^n - \mathbb{E}\hat{v}^n)^2] = \text{Var}(\hat{v}^n) = \sigma^2 + \gamma^2\text{Var}(\hat{v}^{n-1})$$

and

$$2\gamma\text{Var}(\hat{v}^{n-1}) \leq (1 + \gamma^2)\text{Var}(\hat{v}^{n-1})$$

since $\gamma^2 - 2\gamma + 1 = (\gamma - 1)^2 \geq 0$ and $\text{Var}(\hat{v}^{n-1}) \geq 0$ also. This completes the proof. □
Due to Proposition 3, we can solve (11) by setting $\frac{df}{d\alpha_n} = 0$ and solving for $\alpha_{n-1}$. This yields an equation

$$(\alpha_{n-1} - 1) \mathbb{E} \left[ (\bar{v}^{n-1} - \mathbb{E}\hat{v}^n)^2 \right] + \alpha_{n-1} \mathbb{E} \left[ (\hat{v}^n - \mathbb{E}\hat{v}^n)^2 \right] + (1 - 2\alpha_{n-1}) Cov (\bar{v}^{n-1}, \hat{v}^n) = 0$$

whence we obtain

$$\alpha_{n-1} = \frac{\mathbb{E} \left[ (\bar{v}^{n-1} - \mathbb{E}\hat{v}^n)^2 \right] - Cov (\bar{v}^{n-1}, \hat{v}^n)}{\mathbb{E} \left[ (\bar{v}^{n-1} - \mathbb{E}\hat{v}^n)^2 \right] + \mathbb{E} \left[ (\hat{v}^n - \mathbb{E}\hat{v}^n)^2 \right] - 2 Cov (\bar{v}^{n-1}, \hat{v}^n)}.$$ (15)

We now present our main result, which gives an explicit formula for (15).

**Theorem 3.** Assuming that $\alpha_0$ is given, the optimal time-$n$ stepsize can be computed using the formula

$$\alpha_{n-1} = \frac{(1 - \gamma) \lambda^{n-1} \sigma^2 + (1 - (1 - \gamma) \delta^{n-1})^2 c^2}{(1 - \gamma)^2 \lambda^{n-1} \sigma^2 + (1 - (1 - \gamma) \delta^{n-1})^2 c^2 + \sigma^2} \quad n = 2, 3, \ldots$$ (16)

where $\delta^{n-1}$ and $\lambda^{n-1}$ are as in Proposition 1.

**Proof:** We compute each expectation in (15). First, observe that

$$\mathbb{E} \left[ (\hat{v}^n - \mathbb{E}\hat{v}^n)^2 \right] = Var (\hat{v}^n) = Var (\hat{v}^n + \gamma \bar{v}^{n-1}) = (1 + \gamma^2 \lambda^{n-1}) \sigma^2$$

using the independence of $\hat{v}^n$ and $\bar{v}^{n-1}$ together with Proposition 1. Now, recall from (14) that

$$\mathbb{E} \left[ (\bar{v}^{n-1} - \mathbb{E}\hat{v}^n)^2 \right] = Var (\bar{v}^{n-1}) + (\mathbb{E}\bar{v}^{n-1} - \mathbb{E}\hat{v}^n)^2.$$ 

By Proposition 1,

$$Var (\bar{v}^{n-1}) = \lambda^{n-1} \sigma^2.$$ 

and

$$\mathbb{E}\hat{v}^n - \mathbb{E}\bar{v}^{n-1} = c + \gamma \delta^{n-1} c - \delta^{n-1} c = (1 - (1 - \gamma) \delta^{n-1}) c$$

represents the bias of $\bar{v}^{n-1}$ in predicting $\hat{v}^n$. Thus,

$$\mathbb{E} \left[ (\bar{v}^{n-1} - \mathbb{E}\hat{v}^n)^2 \right] = \lambda^{n-1} \sigma^2 + (1 - (1 - \gamma) \delta^{n-1})^2 c^2.$$ 

Finally, it follows from (13) and Proposition 1 that

$$Cov (\bar{v}^{n-1}, \hat{v}^n) = \gamma Var (\bar{v}^{n-1}) = \gamma \lambda^{n-1} \sigma^2.$$ 

Substituting all of these expressions into (15) completes the proof. \qed
Corollary 2. For all $n$, $\alpha_{n-1} \in [0, 1]$.

Proof: The positivity of $\alpha_{n-1}$ is obvious from (16), where both the numerator and denominator are sums of positive terms (it can easily be seen that $\lambda^{n-1} \geq 0$ for all $n$). To show that $\alpha_{n-1} \leq 1$, first observe that

$$\gamma (1 - \gamma) \lambda^{n-1} \sigma^2 \leq \sigma^2$$

by the result of Proposition 2. From this it can easily be shown that

$$(1 - \gamma) \lambda^{n-1} \sigma^2 \leq (1 - \gamma)^2 \lambda^{n-1} \sigma^2 + \sigma^2,$$

completing the proof. □

The formula given in (16) offers an interesting comparison to the BAKF formula in (9). We see that both the numerator and the denominator of the fraction in (16) include covariance terms. These terms do not appear in the BAKF formula, because the work by George & Powell (2006) assumes independent observations. Secondly, because of the specific structure of our problem, we are able to derive an explicit expression for the bias $E \tilde{v}^{n-1} - E \hat{v}^n$.

We show that our formula behaves correctly in special cases. If the rewards we collect are deterministic, then our estimate $\tilde{v}^n$ is simply adding up the discounted rewards, and should converge to $v^*$ under the optimal stepsize rule. If the process $\hat{v}^n$ is stationary, i.e. $\gamma = 0$, then $\tilde{v}^n$ is simply estimating $c$, and we should be using the known optimal stepsize rule of $\alpha_{n-1} = \frac{1}{n}$.

Corollary 3. If the underlying reward process has zero noise, then $\sigma^2 = 0$ and $\alpha_{n-1} = 1$ for all $n$.

It follows that $\tilde{v}^n = \hat{v}^n$ for all $n$, and

$$\lim_{n \to \infty} \tilde{v}^n = \sum_{i=0}^{\infty} \gamma^i c = \frac{c}{1 - \gamma}.$$

Corollary 4. If the problem is stationary, that is, $\gamma = 0$, then the optimal stepsize is given by $\alpha_{n-1} = \frac{1}{n}$ for all $n$ as long as $\alpha_1 = 1$.

Proof: If $\alpha_1 = 1$ and $\gamma = 0$, then $\hat{v}^n = \bar{c}^n$ and $\tilde{v}^n = (1 - \alpha_{n-1}) \bar{c}^{n-1} + \alpha_{n-1} \bar{c}^n$ for all $n$, and $E \tilde{v}^n = c$ for all $n$, which means that $\delta^n = 1$ for all $n$. Then, (16) reduces to

$$\alpha_{n-1} = \frac{\lambda^{n-1} \sigma^2}{(1 + \lambda^{n-1}) \sigma^2} = \frac{\lambda^{n-1}}{1 + \lambda^{n-1}}.$$
We claim that $\lambda^{n-1} = \frac{1}{n-1}$. It is clearly true that $\lambda^1 = 1$, from which it follows that $\alpha_1 = \frac{1}{2}$. Now suppose that $\alpha_{n-2} = \frac{1}{n-1}$ and $\lambda^{n-2} = \frac{1}{n-2}$. Then,

$$
\lambda^{n-1} = \alpha_{n-2}^2 + (1 - \alpha_{n-2})^2 \lambda^{n-2} = \frac{1}{(n-1)^2} + \frac{n-2}{(n-1)^2} = \frac{n-1}{(n-1)^2} = \frac{1}{n-1}
$$

and $\alpha_{n-1} = \frac{1}{n}$, as required. \hfill \Box

We now analyze the limiting properties of the stepsize rule given by (16). In particular, we show that it satisfies one of the two conditions for convergence that were established in the stochastic approximation literature.

### 3.2 Convergence analysis

Recall that, for deterministic stepsize rules, a stochastic approximation algorithm is provably convergent as long as $\alpha_{n-1} \geq 0$ for all $n$ and

$$
\sum_{n=1}^{\infty} \alpha_{n-1} = \infty, \quad \sum_{n=1}^{\infty} \alpha_{n-1}^2 < \infty.
$$

We show that our stepsize satisfies the first condition by establishing a lower bound on $\alpha_{n-1}$.

**Proposition 4.** For all $n \geq 1$, $\alpha_{n-1} \geq \frac{1 - \gamma}{n}$.

**Proof:** We use an inductive argument to show that

$$
\lambda^{n-1} \geq \frac{1}{n-1} \implies \alpha_{n-1} \geq \frac{1 - \gamma}{n} \implies \lambda^n \geq \frac{1}{n}
$$

for all $n > 1$. Assuming $\alpha_0 = 1$, we have $\lambda^1 = 1$ by definition. Suppose now that $\lambda^{n-1} \geq \frac{1}{n-1}$ for some $n > 1$. We rewrite (16) as

$$
\alpha_{n-1} = \frac{1 - \gamma}{n} - \frac{1 - \gamma}{n} + \frac{(1 - \gamma) \lambda^{n-1} \sigma^2 + (1 - (1 - \gamma) \delta^{n-1})^2 \sigma^2}{(1 - \gamma)^2 \lambda^{n-1} \sigma^2 + (1 - (1 - \gamma) \delta^{n-1})^2 \sigma^2 + \sigma^2}
$$

$$
= \frac{1 - \gamma}{n} + A^{n-1},
$$

where

$$
A^{n-1} = \frac{n (1 - \gamma) \lambda^{n-1} \sigma^2 + (n - (1 - \gamma)) (1 - (1 - \gamma) \delta^{n-1})^2 \sigma^2 - (1 - \gamma)^3 \lambda^{n-1} \sigma^2 - (1 - \gamma) \sigma^2}{n (1 - \gamma)^2 \lambda^{n-1} \sigma^2 + n (1 - (1 - \gamma) \delta^{n-1})^2 \sigma^2 + n \sigma^2}.
$$

The denominator of $A^{n-1}$ is clearly positive. To show that the numerator is positive as well, it suffices to show that

$$
n (1 - \gamma) \lambda^{n-1} \sigma^2 \geq (1 - \gamma)^3 \lambda^{n-1} \sigma^2 + (1 - \gamma) \sigma^2.
$$
Because $1 - \gamma \geq (1 - \gamma)^3$ for $\gamma \in (0, 1)$, it remains to show that
\[(n - 1) (1 - \gamma) \lambda^{n-1} \sigma^2 \geq (1 - \gamma) \sigma^2,
\]
but this holds because $\lambda^{n-1} \geq \frac{1}{n-1}$ by the inductive hypothesis. Thus, $\alpha_{n-1} \geq \frac{1-\gamma}{n}$.

Using the result that $\alpha_{n-1} \geq \frac{1-\gamma}{n}$, we show that $\lambda^n \geq \frac{1}{n}$. Let us write
\[
\lambda^{n-1} = \frac{1}{n-1} + L^{n-1},
\]
\[
\alpha_{n-1} = \frac{1-\gamma}{n} + M^{n-1},
\]
where $L^{n-1}, M^{n-1} \geq 0$. Substituting these expressions into the definition of $\lambda^n$, we obtain
\[
\lambda^n = \left( \frac{1-\gamma}{n} + M^{n-1} \right)^2 + \left( 1 - (1 - \gamma) \left( \frac{1-\gamma}{n} + M^{n-1} \right) \right)^2 \left( \frac{1}{n-1} + L^{n-1} \right).
\]
This expression can be rewritten as
\[
\lambda^n = \frac{(1-\gamma)^2}{n^2} + 2 \frac{1-\gamma}{n} M^{n-1} + (M^{n-1})^2
\]
\[
+ \left( \frac{n - (1-\gamma)^2}{n} - (1 - \gamma) M^{n-1} \right)^2 \frac{1}{n-1} + L^{n-1} G^{n-1},
\]
where $G^{n-1} \geq 0$. Since $n - 1 \leq n - (1-\gamma)^2$, it follows that $\frac{1}{n-1} \geq \frac{1}{n - (1-\gamma)^2}$ and
\[
\lambda^n \geq \frac{1}{n} + \frac{n}{n - (1-\gamma)^2} (M^{n-1})^2 + L^{n-1} G^{n-1},
\]
whence $\lambda^n \geq \frac{1}{n}$, as required. \(\square\)

From Proposition 4, it follows that
\[
\sum_{n=1}^{\infty} \alpha_{n-1} \geq (1 - \gamma) \sum_{n=1}^{\infty} \frac{1}{n} = \infty,
\]
satisfying one of the conditions for convergence. As long as the first condition holds, the second condition $\sum_{n=1}^{\infty} \alpha_{n-1}^2 < \infty$ can often be relaxed (see Kushner & Yin, 1997) to the simple requirement that $\alpha_{n-1} \to 0$. We prove that this condition holds for our stepsize formula. First, we show that the bias term in the stepsize formula converges to zero.

**Proposition 5.** $\lim_{n \to \infty} \delta^n = \frac{1}{1-\gamma}$. \\

16
**Proof:** We first show that the sequence \((\delta^n)_{n=1}^\infty\) is increasing. Recall from Proposition 2 that 
\((1 - \gamma) \delta^{n-1} \leq 1\). It follows that 
\((1 - \gamma) \alpha_{n-1} \delta^{n-1} + \delta^{n-1} \leq \alpha_{n-1} + \delta^{n-1}\), whence
\[
\delta^{n-1} \leq \alpha_{n-1} + (1 - (1 - \gamma) \alpha_{n-1}) \delta^{n-1} = \delta^n,
\]
which means that \((\delta^n)\) is increasing. Since this sequence is also bounded by Proposition 2, it has a limit \(\delta^* \leq \frac{1}{1 - \gamma}\).

Suppose now that \(\delta^* < \frac{1}{1 - \gamma}\). We rewrite the definition of \(\delta^n\) as
\[
(\delta^n - \delta^{n-1}) + \delta^{n-1} = \alpha_{n-1} + (1 - (1 - \gamma) \alpha_{n-1}) \delta^{n-1}.
\]
Subtracting \(\delta^{n-1}\) from both sides yields
\[
(\delta^n - \delta^{n-1}) = \alpha_{n-1} (1 - (1 - \gamma) \delta^{n-1}).
\]
The left-hand side converges to zero as \(n \to \infty\). On the right-hand side, if \(\delta^* < \frac{1}{1 - \gamma}\), then 
\[1 - (1 - \gamma) \delta^{n-1} \to 1 - (1 - \gamma) \delta^* > 0.\] It then follows that \(\alpha_{n-1} \to 0\). However, we can see from (16) that this is impossible if \(\delta^* < \frac{1}{1 - \gamma}\) because, in the limit, both the numerator and denominator will contain the strictly positive term \((1 - (1 - \gamma) \delta^*)^2\). All other terms in both the numerator and denominator of (16) are positive. Therefore, it must be the case that \(\delta^* = \frac{1}{1 - \gamma}\). \(\square\)

**Theorem 4.** \(\lim_{n \to \infty} \alpha_n = 0\).

**Proof:** It is enough to show that \(\lambda^n \to 0\) and apply (16) together with Proposition 5. We show that every convergent subsequence of \(\lambda^n\) must converge to zero using a proof by contradiction.

First, suppose that \(n_k\) is a subsequence satisfying \(\lim_{k \to \infty} \lambda^{n_k} = \ell\). Combining this with Proposition 5, we return to (16) and find
\[
\lim_{k \to \infty} \alpha_{n_k} = \frac{(1 - \gamma) \ell}{(1 - \gamma)^2 \ell + 1}.
\]
We then return to Proposition 1 and derive
\[
\lim_{k \to \infty} \lambda^{n_k+1} = \left[\frac{(1 - \gamma) \ell}{(1 - \gamma)^2 \ell + 1}\right]^2 + \left[1 - \frac{(1 - \gamma)^2 \ell}{(1 - \gamma)^2 \ell + 1}\right]^2 \ell
\]
\[
= \frac{(1 - \gamma)^2 \ell^2}{(1 - \gamma)^2 \ell + 1} + \frac{\ell}{(1 - \gamma)^2 \ell + 1}^2
\]
\[
= \frac{\ell}{(1 - \gamma)^2 \ell + 1}.
\]
(17)
It follows from (17) that, if \( \ell > 0 \), then
\[
\lim_{k \to \infty} \lambda_{n_k+1} < \lim_{k \to \infty} \lambda_{n_k}.
\] (18)

By Proposition 2, we know that the sequence \((\lambda^n)_{n=1}^{\infty}\) is bounded. Therefore, the set of accumulation points for this sequence is closed and bounded. Suppose that
\[
\limsup_{n \to \infty} \lambda^n = \lambda^*
\]
and that \( \lambda^* > 0 \). Let \( n_k \) be a subsequence with \( \lambda_{n_k} \to \lambda^* \). The subsequence \((\lambda^{n_k-1})_{k=1}^{\infty}\) is bounded, and therefore must contain an additional convergent subsequence, which we denote by \( m_k \). Suppose that \( \lim_{k \to \infty} \lambda_{m_k} = \ell \). It must be the case that
\[
\lim_{k \to \infty} \lambda_{m_k+1} = \lim_{k \to \infty} \lambda_{n_k} = \lambda^*.
\]

This implies that \( \ell > 0 \), because otherwise (17) would imply that \( \lambda^* = 0 \). However, it then follows from (18) that \( \lambda^* < \ell \). This is impossible, because we took \( \lambda^* \) to be the largest accumulation point of the sequence \((\lambda^n)_{n=1}^{\infty}\). It must therefore be the case that
\[
\limsup_{n \to \infty} \lambda^n = 0,
\]
whence \( \lambda^n \to 0 \), as required. \( \square \)

It follows immediately from these results that \( \tilde{v}^n \to v^* \) in \( L^2 \) and in probability. Observe that
\[
E \left[ (\tilde{v}^n - v^*)^2 \right] = E \left[ (\tilde{v}^n - E\tilde{v}^n + E\tilde{v}^n - v^*)^2 \right] \\
= Var(\tilde{v}^n) + (E\tilde{v}^n - v^*)^2 \\
= \lambda^n \sigma^2 + \left( \delta^n c - \frac{c}{1 - \gamma} \right)^2.
\] (19)

Together, Proposition 5 and Theorem 4 imply that (19) vanishes to zero as \( n \to \infty \). We defer to Kushner & Yin (1997) for a discussion of situations where \( \alpha_{n-1} \to 0 \) also implies a.s. convergence.

### 3.3 Algorithmic procedure for unknown \( c \)

Suppose that the mean and variance of \( \tilde{v}^n \) are unknown, but the observations themselves can be collected, e.g., through black-box simulation of a system in steady state. We can form estimates of these parameters and plug these into the expression for the optimal stepsizes (this is known
as the plug-in principle; see e.g. Bickel & Doksum, 2001). Both parameters can be estimated by smoothing on the observations $\hat{c}^n$. Define

$$\bar{c}^n = (1 - \nu_{n-1}) \bar{c}^{n-1} + \nu_{n-1} \hat{c}^n$$  \hspace{1cm} (20)

to represent our estimate of the mean. The stepsize $\nu_{n-1}$ is chosen according to some deterministic stepsize rule. Similarly, let

$$(\bar{\sigma}^n)^2 = (1 - \nu_{n-1}) (\sigma^{n-1})^2 + \nu_{n-1} (\hat{c}^n - \bar{c}^{n-1})^2$$  \hspace{1cm} (21)

be our estimate of $\text{Var}(\hat{c}^n)$. Finally, (16) becomes

$$\alpha_{n-1} = \frac{(1 - \gamma) \lambda^{n-1} (\bar{\sigma}^n)^2 + (1 - (1 - \gamma) \delta^{n-1})^2 (\bar{c}^n)^2}{(1 - \gamma)^2 \lambda^{n-1} (\sigma^n)^2 + (1 - (1 - \gamma) \delta^{n-1})^2 (\bar{c}^n)^2 + (\bar{\sigma}^n)^2} \hspace{1cm} n = 2, 3, ...$$  \hspace{1cm} (22)

where $\delta^{n-1}$, $\lambda^{n-1}$ are computed the same way as before.

At first glance, it appears that we run into the same problem as the BAKF rule in (9), which also requires a second stepsize $\nu_{n-1}$ to estimate an unknown parameter. However, in the case of BAKF, the unknown parameter is the bias $\beta^n = \theta^n - \mathbf{E}(\check{\theta}^{n-1})$, which evolves with $n$. The bias is inherently difficult to estimate; if we could know it exactly, then we would have the true unknown means. In the single-state, single-action problem, the bias can be computed exactly, and the second stepsize $\nu_{n-1}$ is only required to estimate the parameters of the distribution of the one-period reward $\hat{c}^n$. The one-period reward is stationary, and can be straightforwardly estimated from the random rewards collected in each time period.

The stationarity of $\hat{c}^n$ suggests that we should use $\nu_{n-1} = 1/n$. However, the true significance of (22) is that it can be easily extended to a general MDP with many states and actions. In this setting, the one-period reward $\hat{c}^n (S^n, a^n, W^{n+1})$ depends on the current state and the action (or the policy). The policy is changing over the iterations, which also changes the probability we are in each state. For this reason, $\hat{c}^n$ would come from a nonstationary sequence. However, as we stated in the beginning of Section 2, the basic value-iteration update of (2) eventually converges to an optimal policy, meaning that the expected one-period reward earned in a state converges to a system-wide constant $\bar{c}$. We can estimate $\bar{c}$ using a slightly modified version of (20), given by

$$\bar{c}^n = (1 - \nu_{n-1}) \bar{c}^{n-1} + \nu_{n-1} \hat{c}^n \left( S^n, a^n, W^{n+1} \right).$$ \hspace{1cm} (23)

Similarly, we estimate $(\bar{\sigma}^n)^2$ using (21) by using $\hat{c}^n = \hat{c}^n \left( S^n, a^n, W^{n+1} \right)$. We emphasize that even for multiple states and actions, we still compute a single, system-wide value for $\bar{c}^n$ and $(\bar{\sigma}^n)^2$ using
equations (20) and (21), respectively, producing a single stepsize, rather than one that depends on state (or state-action pair). Thus, while there are multiple states and actions, we are continuing to view it in aggregate as a single system. If we are using approximate value iteration, we update the value function estimates using equation (2) and (3).

In this general setting, we suggest using a constant stepsize in (23), e.g. \( \nu_{n-1} = 0.2 \), to avoid giving equal weight to early observations, which are taken in the transient period before the MDP has reached steady state, while the probability of being in each state is still changing due to changes in the policy. Our numerical work suggests the results are not very sensitive to the choice of \( \nu_{n-1} \).

### 3.4 Extension to finite horizon

While it is possible to solve finite horizons using the same algorithmic strategy, we observe that optimal stepsizes vary systematically as a function of the number of time periods to the end of horizon. The best stepsize for states at the end of the horizon is very close to \( 1/n \), because we do not face the need to sum rewards over a horizon. Optimal stepsizes then increase as we move closer to the first time period.

We can capture this behavior using a finite horizon version of our single-state, single-action problem. In this setting, approximate value iteration is replaced with approximate dynamic programming. Equations (4) and (5) become

\[
\hat{v}_t^n = \hat{c}_t^n + \gamma \bar{v}_{t+1}^{n-1} \tag{24}
\]

\[
\bar{v}_t^n = (1 - \alpha_{n-1,t}) \bar{v}_{t}^{n-1} + \alpha_{n-1,t} \hat{v}_t^n. \tag{25}
\]

These equations are solved for \( t = 1, \ldots, T - 1 \) in each time step \( n \). We assume that \( \bar{v}_T^n = 0 \) for all \( n \), and that the observations \( \hat{c}_t^n \) are independent and identically distributed for all \( n \) and \( t \).

Our analysis can easily be extended to this setting. First, we can obtain expressions for the expected value and variance of \( \bar{v}_t^n \) that generalize our derivations of \( \delta^n \) and \( \lambda^n \) in Section 3.1. The following proposition describes these expressions.

**Proposition 6.** For \( t = 1, \ldots, T - 1 \), define

\[
\delta_t^n = \begin{cases} 
\alpha_{0,t} & n = 1 \\
\frac{\alpha_{0,t} \delta_{t+1}^{n-1}}{1 + \gamma \delta_{t+1}^{n-1}} + (1 - \alpha_{n-1,t}) \delta_t^{n-1} & n > 1
\end{cases}
\]
Also, for $t, t' = 1, \ldots, T - 1$, let

\[
\lambda_{t,t'}^n = \begin{cases} 
\alpha_{0,t}^2 I_{\{t=t'\}} \\
\alpha_{n-1,t}^2 I_{\{t=t'\}} + J_{t,t'}^{n-1} + K_{t,t'}^{n-1} + L_{t,t'}^{n-1} + M_{t,t'}^{n-1} & n > 1
\end{cases}
\]

where

\[
J_{t,t'}^{n-1} = (1 - \alpha_{n-1,t}) (1 - \alpha_{n-1,t'}) \lambda_{t,t'}^{n-1},
\]

\[
K_{t,t'}^{n-1} = \gamma (1 - \alpha_{n-1,t}) \alpha_{n-1,t'} \lambda_{t,t'+1}^{n-1},
\]

\[
L_{t,t'}^{n-1} = \gamma \alpha_{n-1,t} (1 - \alpha_{n-1,t'}) \lambda_{t+1,t'}^{n-1},
\]

\[
M_{t,t'}^{n-1} = \gamma^2 \alpha_{n-1,t} \alpha_{n-1,t'} \lambda_{t+1,t'+1}^{n-1}.
\]

Then, $\mathbb{E}(\bar{v}_t^n) = \delta_t^n c$ and $\text{Cov}(\bar{v}_t^n, \bar{v}_{t'}^n) = \lambda_{t,t'}^n \sigma^2$.

The proof uses the same logic as the proof of Proposition 1. We can think of $\lambda^n$ as a symmetric matrix that can be updated recursively using the elements of $\lambda_{n-1}^{n-1}$. The matrix starts out diagonal, and as $n$ increases, the covariances gradually expand from the main diagonal outward. Next, we can repeat the analysis of Section 3.1 to solve

\[
\min_{\alpha_{n-1,t} \in [0,1]} \mathbb{E} \left[ (\bar{v}_t^n (\alpha_{n-1,t}) - \mathbb{E}(\bar{v}_t^n))^2 \right].
\]

The next result gives the solution.

**Theorem 5.** If $\alpha_{0,t}$ is given, the optimal stepsize for time $t$ at iteration $n$ is given by

\[
\alpha_{n-1,t} = \frac{\left( \lambda_{t,t}^{n-1} - \gamma \lambda_{t,t+1}^{n-1} \right) \sigma^2 + (1 - \delta_{t}^{n-1} + \gamma \delta_{t+1}^{n-1})^2 c^2}{\left( \lambda_{t,t}^{n-1} - 2 \gamma \lambda_{t,t+1}^{n-1} + \gamma^2 \lambda_{t+1,t+1}^{n-1} \right) \sigma^2 + (1 - \delta_{t}^{n-1} + \gamma \delta_{t+1}^{n-1})^2 c^2 + \sigma^2}.
\]

In the infinite-horizon case, this reduces to our original formula in (16). The finite-horizon formula requires us to store more parameters in the form of a matrix $\lambda^n$, but it is also able to vary the stepsize depending on the value of $t$. If $c$ and $\sigma^2$ are unknown, we can adapt the approximation procedure outlined in Section 3.3, and replace the unknown values in (26) with $\bar{c}^n$ and $(\bar{\sigma}^n)^2$.

### 4 Experimental study: one state, one action

We first study the performance of our stepsize rule on an instance of the single-state, single-action problem. This allows us to obtain insights into the sensitivity of performance with respect to
different problem parameters. As in Section 2, we let the mean reward $c = 1$. The discount factor was chosen to be $\gamma = 0.9$, thus the optimal value for this problem is $V^* = 10$. Four different stepsize rules were implemented; we briefly describe them as follows.

**Optimal stepsize for approximate value iteration (OSAVI).** We use the approximate version of the optimal stepsize, given by (22). The secondary stepsize $\nu_{n-1}$ was set to 0.2.

**BAKF.** We use the approximate version of BAKF, given by George & Powell (2006, Fig. 4). The stepsize rule is similar to (9), but uses a secondary stepsize rule $\tilde{\nu}_{n-1} = 0.05$ to estimate $\beta^n$, similar to how $c$ is estimated in (22).

**McClain’s rule.** McClain’s stepsize formula is given by

$$\alpha_n = \begin{cases} 
\frac{1}{\alpha_{n-1}} & \text{if } n = 1 \\
\frac{\alpha_{n-1}}{1+\alpha_{n-1} - \bar{\alpha}} & \text{otherwise},
\end{cases}$$

where $\bar{\alpha}$ is a tunable parameter. This stepsize behaves like the $1/n$ rule in early iterations, but quickly converges to the limit point $\bar{\alpha}$. This tends to happen within approximately 10 iterations. For our experiments, we used $\bar{\alpha} = 0.1$; the issue of tuning $\bar{\alpha}$ is discussed in Section 4.2. McClain’s rule should be viewed as a slightly more sophisticated version of a constant stepsize.

**Harmonic stepsize.** This deterministic rule is given by $\alpha_{n-1} = \frac{a}{a + n}$, where $a > 0$ is a tunable parameter. A value of $a = 10$ yielded good performance for our choice of problem parameters. However, the harmonic stepsize is sensitive to the choice of the tunable parameter $a$, which is highly problem dependent. If we expect good convergence in a few hundred iterations, $a$ on the order of 5 or 10 may work quite well. On the other hand, if we anticipate running our algorithm millions of iterations (which is not uncommon in reinforcement learning), we might choose $a$ on the order of 10,000 or higher. This issue is discussed further in Section 4.2.

We also considered the stochastic gradient stepsize of Benveniste et al. (1990) and the Delta-Bar-Delta rule of Sutton (1992), as well as the polynomial stepsize $\alpha_{n-1} = 1/n^\beta$. However, these stepsizes consistently underperformed the four rules listed above, and are omitted from the subsequent analysis. The constant rule $\alpha_{n-1} = \bar{\alpha}$ yielded results very similar to McClain’s rule, and is also omitted.
4.1 Numerical evaluation of stepsize rules

Figure 3 shows the value of the objective function in (11) achieved by each stepsize rule over $10^4$ iterations. The OSAVI rule consistently achieves the best performance (lowest objective value). However, the harmonic stepsize, when properly tuned, performs comparably. The BAKF and McClain rules level off around an objective value of $10^{-2}$. Each data point in Figure 3 is an average over $10^4$ simulations.

By minimizing the objective function in (11), we strike a balance between quickly approaching the optimal value $V^* = 10$ and achieving a stable estimate of this quantity. Figure 4(a) shows the effect of each stepsize rule on the value function approximation $\tilde{v}^n$ for a single sample path. We see that the BAKF rule makes the value function approximation approach $V^*$ more quickly than the other stepsizes. However, the BAKF estimate is unstable, with a large amount of variation around $V^*$. McClain’s rule also results in a high amount of volatility, while approaching $V^*$ slowly. OSAVI approaches $V^*$ more quickly than the harmonic stepsize, but both result in very stable estimates.

Figure 4(b) shows how the actual stepsizes $\alpha_{n-1}$ evolve over time under the different stepsize rules, under the same sample path shown in Figure 4(a). In the early iterations, OSAVI sharply decreases the magnitude of $\alpha_{n-1}$, then corrects for this behaviour. The stepsizes climb back up,
and eventually follow the same general trajectory as under the harmonic rule for a tuned value of $a$. Overall, the BAKF rule yields higher values of $\alpha_{n-1}$ than OSAVI. However, under the BAKF rule, the stepsizes never decline to zero. Instead, they oscillate sharply in the later iterations. This issue is due to our use of a constant stepsize in estimating the bias term in the BAKF rule. We discuss this issue further in Section 4.2.

### 4.2 Discussion of tunable parameters

Each of the stepsize rules considered in Section 4 uses a tunable parameter. The sensitivity of each rule to the proper choice of tunable parameter is an important issue. We begin by considering the approximate BAKF rule, which uses a secondary stepsize $\bar{\nu}_{n-1}$ to estimate the bias $\beta^n$. Figure 5(a) shows the effect of varying $\bar{\nu}_{n-1}$ on the objective value achieved by BAKF (with the optimal stepsize shown for comparison).

We see that, when we use a constant value for the secondary stepsize (e.g. $\bar{\nu}_{n-1} = 0.05$), there is a clear tradeoff between performance in the early and late iterations. Smaller values of $\bar{\nu}_{n-1}$ result in better performance in the long run (the objective value achieved by BAKF plateaus at a lower level), but worse performance in the short run. In terms of the quality of our approximation of $V^*$, smaller constants cause slower convergence, but more stable estimates.

It is also necessary to note one special case, where $\bar{\nu}_{n-1} = 1/n$. If we use the $1/n$ rule for the secondary stepsize, the objective value achieved by BAKF declines to zero in the long run. This is
not the case when we use a constant stepsize. Furthermore, the use of the $1/n$ rule produces very close performance to that of OSAVI. However, in a general MDP, where there are many different rewards, a constant stepsize may be better able to handle the transient phase of the MDP. For this reason, we focus primarily on constant values of $\bar{\nu}_{n-1}$ in this study.

Even with a declining secondary stepsize, the BAKF rule is outperformed by OSAVI with a simple constant secondary stepsize of $\nu_{n-1} = 0.2$. The results for different values of $\bar{\nu}_{n-1}$ indicate that BAKF is quite sensitive to the choice of $\bar{\nu}_{n-1}$.

Figure 5(b) suggests that OSAVI is relatively insensitive to the choice of secondary stepsize. The lines in Figure 5(b) represent the performance of OSAVI for values of $\nu_{n-1}$ ranging from 0.05 to as high as 0.5. We see that these changes have a much smaller effect on the performance of OSAVI than varying $\bar{\nu}_{n-1}$ had on the BAKF rule. Very small values of $\nu_{n-1}$, such as 0.05, do yield slightly poorer performance, but there is little difference between 0.2 and 0.5. Furthermore, the objective value achieved by OSAVI declines to zero for each constant value of $\nu_{n-1}$, whereas BAKF always levels off under a constant secondary stepsize. We conclude that OSAVI is more robust than BAKF, and requires less tuning of the secondary stepsize.

Figure 6(a) shows the sensitivity of McClain’s rule to the choice of tunable parameter $\bar{\alpha}$. The effect is very similar to the effect of using different constant values of $\bar{\nu}_{n-1}$ in Figure 5(a). Smaller values of $\bar{\alpha}$ give better (more stable) late-horizon performance and worse (slower) early-horizon performance.
The harmonic rule is analyzed in Figure 6(b). We see that $a = 10$ is clearly the optimal choice for this problem, with the particular parameter values (variance and discount factor) that we have chosen. Larger values of $a$ are consistently worse, and smaller values are only effective in the very early iterations. However, $a = 10$ yields very good performance, the best out of all the competing stepsize rules.

In fact, it is possible to tune the harmonic rule to perform competitively against OSAVI. However, the best value of $a$ is highly problem-dependent. Figure 7(a) shows that $a = 10$ continues to perform well when $\sigma^2$ is increased to 4, and even achieves a slightly lower objective value than the approximate optimal rule in the later iterations, although OSAVI performs noticeably better in the early iterations. However, Figure 7(b) shows that $a = 100$ becomes the best value when the discount factor $\gamma$ is increased to 0.99. The optimal rule has not been retuned in Figure 7; all results shown are for $\nu_{n-1} = 0.2$.

We conclude based on Figures 6(b) and 7 that the best choice of $a$ in the harmonic stepsize rule is very sensitive to the parameters of the problem, and that the best choice of $a$ for one problem setting can perform very poorly for a different problem. By contrast, Figure 5(b) shows that OSAVI is relatively insensitive to its tunable parameter. A simple value of $\nu_{n-1} = 0.2$ yields good results in all of the settings considered. We claim that OSAVI is a robust alternative to several leading stepsize rules.
Figure 7: Sensitivity of the harmonic stepsize rule in different problem settings.

5 Experimental study: general MDP

We also tested the OSAVI rule on a more general MDP with 100 states and 10 actions per state. We deliberately created a problem where we can compute an optimal solution by discretizing the problem and solving it using standard backward dynamic programming methods. This gives us a rigorous benchmark for evaluating the quality of the solution.

The MDP was generated randomly according to the procedure used by Even-Dar & Mansour (2003). For every combination $s, s'$ of states and for every action $a$, we generated a number $b_{s,s',a} \sim U[0,1]$. The probability of making a transition from state $s$ to state $s'$ upon choosing action $a$ was set to be $\frac{b_{s,s',a}}{\sum_{s''} b_{s,s'',a}}$. Furthermore, we generated a number $c_{s,a} \sim U[0,10]$ representing the one-period reward for taking action $a$ in state $s$. We considered both finite- and infinite-horizon settings.

5.1 Infinite-horizon setting

We run approximate value iteration by letting $\bar{V}^0(s) = 0$ for all $s$ and updating the value of being in state $s^n$ using the equations

\begin{align}
\hat{v}^n &= \max_a c_{s^n,a} + \gamma \sum_{s'} P(s'|s^n,a) \bar{V}^{n-1}(s') \\
\bar{V}^n(s^n) &= (1 - \alpha_{n-1}) \hat{v}^n(s^n) + \alpha_{n-1} \hat{v}^n,
\end{align}

(27) (28)
where $\alpha_{n-1}$ is chosen according to some stepsize rule. Observe that, if we let $a^n$ be the action that achieves the maximum in the definition of $\hat{v}^n$ and simulate the next state according to $P(s' | s^n, a^n)$, the stepsize rule will change the policy used to make decisions, and thus the probability of being in a state in future iterations. To avoid this complication, we use offline learning, whereby the equations above are used to update the target policy, while the actual next state $s^{n+1}$ is chosen uniformly at random.

This MDP is still sufficiently simple to enable us to find the optimal value $V(s)$ of being in state $s$ using value iteration. Thus, a natural measure of the quality of the value function approximation after $N$ iterations is the mean squared error (MSE), given by

$$\frac{1}{|S|} \sum_s (V(s) - V^N(s))^2.$$  

In our experiments, this quantity is averaged over $10^4$ sample paths. We can estimate the standard error by dividing these sample paths in groups of 500 to obtain approximately normal estimates of the average MSE, then computing the sample standard deviation of the group means. These numbers are omitted from the subsequent discussion because they tended to be two orders of magnitude smaller than the differences in MSE between stepsize rules.

We compared the same stepsize rules listed in Section 4: McClain’s rule with $\bar{\alpha} = 0.1$, the harmonic rule with $a = 10$ and $a = 100$ (these are the tuned values that were found in Section 4.2 to work best for $\gamma = 0.9$ and $\gamma = 0.99$, respectively), the BAKF rule of George & Powell (2006), and OSAVI. For these experiments, we made the stepsizes state-dependent in order to achieve quicker

![Figure 8: Average MSE achieved by each stepsize rule for various discount factors in the 100-state MDP.](image)
convergence. For example, the harmonic rule is given by
\[ \alpha_{n-1}(s) = \frac{s_{n-1}}{s_{n-1} + N_s^n} \]
where \( N_s^n \) is the number of times state \( s \) was visited in \( n \) iterations. The parameters \( \delta^n, \lambda^n \) and \( \zeta^n \) used by OSAVI and BAKF were also chosen to be state-dependent.

Figure 8 shows the average MSE achieved by each stepsize rule over time, up to 10000 iterations. For \( \gamma = 0.9 \), OSAVI generally outperforms McClain’s rule and the harmonic rule with \( a = 10 \), while the other two stepsize rules are competitive early on. After about 3000 iterations, the BAKF rule begins to yield much slower improvement in the MSE. After about 4500 iterations, OSAVI pulls ahead of harmonic with \( a = 100 \) as well. For higher \( \gamma \), we see roughly the same behaviour, except OSAVI pulls ahead of the competition sooner (around 3000 iterations). We conclude that OSAVI yields generally competitive performance.

5.2 Finite-horizon setting

Finite-horizon problems introduce the dimension that the relative size of the learning bias versus the noise in the contribution depends on the time period. As a result, the optimal stepsize behaviour changes with time. In the finite-horizon case, we run approximate dynamic programming by computing

\[
\hat{v}_t^n = \max_a c_{s_t^n, a} + \gamma \sum_{s'} P(s' | s_t^n, a) \hat{V}_{t+1}^{n-1}(s')
\]

\[
\hat{V}_t^n(s_t^n) = (1 - \alpha_{n-1,t}) \hat{V}_t^{n-1}(s_t^n) + \alpha_{n-1} \hat{v}_t^n
\]

for \( t = 1, \ldots, T - 1 \). The true value \( V_t(s) \) of being in state \( s \) at time \( t \) can be found using backward dynamic programming. The performance measure is the MSE, given by

\[
\frac{1}{|S|} \frac{1}{T} \sum_{t=1}^{T} \sum_s \left( V_t(s) - V_t^N(s) \right)^2,
\]

now computed over all states and all time periods. Once again, to ensure that our stepsize rule does not change the probability of visiting a state, we make transitions by choosing a state uniformly at random. The time horizon used was \( T = 20 \), and the discount factor used was \( \gamma = 0.99 \).

We compared the approximate version of the finite-horizon OSAVI rule from (26) to the harmonic rule \( \alpha_{n-1,t} = \frac{a}{a+n} \) with \( a = 10 \) and \( a = 100 \). This rule was found to be the most competitive (when properly tuned) in the previous experiments. As before, all stepsizes were made to be state-dependent. Figure 9(a) shows the average MSE of each stepsize rule. We see that the harmonic
rule performs better when \( a = 100 \), but OSAVI yields better performance if \( a = 10 \). Once again, the harmonic rule is quite sensitive to the choice of \( a \).

Figure 9(b) shows how the magnitude of the stepsize \( \alpha_{n-1,t} \) produced by the OSAVI formula changes with the number of times we visit a state for different times \( t \). When \( t = 19 \), OSAVI is identical to the \( 1/n \) rule, as we would expect, since this is the last time in the horizon. We assume \( V_{20}(s) = 0 \) for all \( s \), so the observations \( \hat{v}_{19} \) are stationary, and Corollary 4 applies. For values of \( t \) earlier in the time horizon, the optimal stepsize steadily increases, with the largest values of \( \alpha_{n-1,t} \) being for \( t = 1 \). It takes a long time for our observations to propagate backward across the time horizon, and so we need larger stepsizes at time \( t = 1 \) to ensure that these observations have an effect. We note that, for earlier time periods, OSAVI goes through a period of exploration before settling on a curve which can be closely approximated by \( \frac{a}{a+n} \) for a suitably calibrated choice of \( a \).

For the finite-horizon problem, \( a \) would be different for each time period. Thus, we conclude that finite-horizon OSAVI is more robust than the harmonic rule.

6 Experimental study: ADP for a continuous inventory problem

The last part of our experimental study demonstrates how OSAVI can be used in conjunction with ADP on a problem where the state space is continuous. We present a stylized inventory problem where a generic resource can be bought and sold on the spot market, and held in inventory in the interim. The basic structure of our problem appears in applications in finance (Nascimento &
Powell, 2010), energy (Löhndorf & Minner, 2010), inventory control (Powell & Simão, 2009), and water reservoir management (Cervellera et al., 2006). We deliberately abstract ourselves from any particular setting, as we wish to keep the focus on the stepsize rule and test it in a generic setting for which ADP is required.

The state variable of the generic inventory problem contains two dimensions. Let $S_t = (R_t, P_t)$, where $R_t$ denotes the amount of resource currently held in inventory, and $P_t$ denotes the current spot price of the resource. We assume that $R_t$ can take values in the set $\{0, 0.02, 0.04, \ldots, 1\}$, representing a percentage of total inventory capacity. The action $x_t$ represents our decision to buy more inventory (positive values) or sell from our current stock (negative values). We assume that we can buy or sell up to 50% of the total capacity in one time step, again in increments of 2%. Thus, there are up to 50 actions in the problem. The reward $C(P, x) = -E \cdot P \cdot x$ represents the revenue obtained (or cost incurred) after making decision $x$ given a price $P$. Since $x$ is a percentage, the actual amount of inventory bought or sold is given by $E \cdot x$.

While the resource variable $R_t$ is discrete, we assume that the spot price $P_t$ is continuous, and follows a geometric Ornstein-Uhlenbeck (mean reverting) process, a standard price model in finance and other areas. With minor modifications to the problem, $P_t$ could also be changed into an exogenous supply process, which we could draw from to satisfy a demand. The important aspect is that $P_t$ is continuous, which makes it impossible to solve (1) for every state. Furthermore, even for a given state $S_t$, computing the expectation in (1) is difficult, because the transition to the next state $S_{t+1}$ depends on a continuous random variable. For these reasons, we approach the problem using approximate dynamic programming with a discrete value function approximation.

To address the issue of the continuous transition to the next state, we use the post-decision state concept introduced in Van Roy et al. (1997) and discussed extensively by Powell (2007). Given a state $S_t$ and a decision $x_t$, the post-decision state $S^x_t = (R^x_t, P^x_t)$ is given by the equations

$$R^x_t = R_t + x_t,$$
$$P^x_t = P_t.$$

The next pre-decision state $S_{t+1}$ is then obtained by setting $R_{t+1} = R^x_t$ and simulating $P_{t+1}$ from the price process. Given a value function approximation $\hat{V}^{n-1}$, the update $\hat{v}^n_t$ is computed using

$$\hat{v}^n_t = \max_{x_t} C(P^n_t, x_t) + \gamma \hat{V}^{n-1}_t(S^n_{x,t}).$$
This quantity is then used to update the previous post-decision state, that is,
\[ V_{t-1}^{n} (S_{t-1}) = (1 - \alpha_{n-1,t-1}) V_{t-1}^{n-1} (S_{t-1}) + \alpha_{n-1,t-1} \hat{v}_t^n. \]

Thus, we can adaptively improve our value function approximation without computing an expectation.

For our value function approximation, we used a lookup table where the log-price log \( P_t \) was discretized into 34 intervals of width 0.125 between −2 and 2. Thus, the table contained a total of 51 \cdot 34 = 1734 entries. However, while the approximation used a discretized state space, our experiments simulated \( P_t \) using the continuous price process. The price was only discretized during calls to the lookup table. This is an important detail: while we use a discrete approximation, we are still solving the original continuous problem.

We evaluated the performance of each stepsize rule (under a pure exploration behaviour policy where each action \( x_t \) was chosen uniformly at random) by simulating the final VFAs obtained after \( N \) iterations offline. That is, we fixed the final approximation \( V^N \) resulting from a particular stepsize rule and simulated the total infinite-horizon discounted reward obtained by making decisions of the form \( x_t = \arg \max_x C (P_t, x) + \gamma V^N_t (S_t^x) \). This quantity was averaged over \( 2.5 \times 10^4 \) sample paths. Figure 10 reports the quality of the approximation obtained using different stepsize rules in both finite- and infinite-horizon settings. Since our objective is to maximize revenue, higher numbers on the \( y \)-axis represent better quality. As before, we ensure that the choice of stepsize does not affect the target policy by choosing \( R_{t+1} \) uniformly at random and generating log \( P_t \) from a uniform distribution.

Figure 10(a) shows the performance of OSAVI in the infinite-horizon setting. Because of the larger size of the inventory problem, we require several thousand iterations in order to obtain any improvement in the target policy specified by \( V^N \). After 4000 iterations, we find that OSAVI consistently yields the most improvement in the value of the target policy. Analogously to Figure 9(a) in Section 5.2, we also compared OSAVI to the harmonic rule in a finite-horizon setting; the results are shown in Figure 10(b). As in the infinite-horizon setting, several thousand iterations are required before any improvement can be observed, but OSAVI consistently outperforms the best version of harmonic.

Our experiments on the inventory problem offer additional evidence that our new stepsize rule can be applicable to more complex dynamic programming problems, which cannot be solved exactly,
Figure 10: Offline policy values for different stepsize rules in the inventory problem with (a) infinite horizon and (b) finite horizon.

and where additional techniques such as the post-decision state variable are necessary to deal with continuous state spaces and difficult expectations. Even in the streamlined form considered here, the inventory problem features a continuous price variable, and the value of being in a state depends on the behavior of a mean-reverting stochastic differential equation. The fact that OSAVI retains its advantages over other stepsize rules in this setting is an encouraging sign.

7 Conclusion

We have analyzed some important issues related to stepsize selection in approximate value iteration. First, we have derived new rate of convergence results for the popular $1/n$ stepsize rule in the context of a single-state, single-action MDP. In this simple problem, approximate value iteration converges so slowly under the $1/n$ rule as to be virtually unusable for most infinite-horizon applications. This underscores the importance of stepsize selection in general dynamic programming problems.

We have also derived a new stepsize rule that minimizes the squared prediction error of the value function approximation in the single-state, single-action MDP. Our analysis is motivated by earlier work on Kalman filters. However, to our knowledge, our stepsize rule is the first to take into account the covariance between the observation we make of the value of being in a state, and our approximation of that value, a property that is inherent in approximate dynamic programming algorithms. Furthermore, we are able to compute a closed-form expression for the prediction bias in the single-state, single-action case, which considerably simplifies the task of estimating this quantity.
in the general case. We have also outlined a method for extending our stepsize rule to a general MDP setting, both finite- and infinite-horizon.

We have tested our stepsize rule against two leading deterministic rules and one stochastic rule. In the single-state, single-action case, we consistently outperform the other stepsize rules. While some competing rules (particularly the harmonic rule) can be tuned to yield very competitive performance, they are also very sensitive to the choice of tuning parameter. On the other hand, our stepsize rule is robust, displaying little sensitivity to the parameter used to estimate the one-period reward. We also tested our stepsize rule on a general discrete-state MDP, as well as on a more complex ADP problem. We found that OSAVI performs competitively against the other rules in both finite- and infinite-horizon settings.

We conclude that our stepsize rule can be a good alternative to other leading stepsizes. Our conclusion reflects the particular set of experiments that we chose to run. It is important to remember that deterministic stepsizes such as the harmonic rule can be finely tuned to a particular problem, resulting in better performance than the adaptive rule that we present. The strength of our rule, however, is its ability to adjust to the evolution of the value function approximation, as well as its relative lack of sensitivity to tuning.

Acknowledgments

The authors are grateful to the Topology Atlas forum for several very helpful discussions. This research was supported in part by AFOSR contract FA9550-08-1-0195 and ONR contract N00014-07-1-0150 through the Center for Dynamic Data Analysis.

References


### 8 Appendix: Slow convergence of $1/n$: technical details

Setting the stepsize to $\alpha_n = \frac{1}{n+1}$, we know (e.g. from Kushner & Yin, 1997) that the approximation is guaranteed to converge to the optimal value. Let $\bar{v}^0 = 0$ be the initial approximation. We rewrite (5) for time $n+1$ using (4) as

$$\bar{v}^{n+1} - \bar{v}^n = \alpha_n(\hat{v}^{n+1} - \bar{v}^n) = \frac{1}{n+1}(c - (1-\gamma)\bar{v}^n),$$

where the above equations hold for $n \in \mathbb{N}^* = \{0, 1, 2, \ldots\}$. We characterize the slow convergence of approximate value iteration smoothed with a $1/n$ stepsize by bounding $\bar{v}^n$ above and below by

$$1 - (n+1)^{-(1-\gamma)} \leq \frac{\bar{v}^n}{v^*} \leq 1 - \frac{\gamma^2 + \gamma - \frac{1}{n^{-(1-\gamma)}}}{\gamma} - \frac{1 - \gamma}{\gamma} \frac{1}{n}$$

for $n \geq 1$. This bound implies that $\bar{v}^n$ converges particularly slowly for $\gamma$ larger than .8. For example, for $\gamma \geq .9$, this bound tells us that approximate value iteration takes at least $10^{19}$ iterations to reach within 1% of optimal. This is too many iterations for even the fastest implementation.

We approximate the discrete time update equation (29) with a continuous time differential equation in which $\bar{v}^{n+1} - \bar{v}^n$ is approximated by the derivative of $\bar{v}^n$ with respect to $n$. The first step is to extend the definition of $\bar{v}^n$ from the natural numbers onto the positive reals through a piecewise linear interpolation. Define $\bar{\bar{v}}(n)$ as in (7). The right derivative of $\bar{\bar{v}}(n)$ is given by

$$\frac{d^+}{dn} \bar{\bar{v}}(n) = \frac{1}{[n]+1}(c - (1-\gamma)\bar{\bar{v}}([n])).$$

Our strategy is to construct functions $U : \mathbb{R}^2 \to \mathbb{R}$ and $L : \mathbb{R}^2 \to \mathbb{R}$ such that $L(\bar{\bar{v}}(n), n) \leq \frac{d^+}{dn} \bar{\bar{v}}(n) \leq U(\bar{\bar{v}}(n), n)$. Fix any $n_0 \in \mathbb{R}_+$. Then a lower bound for $\bar{\bar{v}}$ on $[n_0, \infty)$ is given by any
solution \( l \) to the differential equation

\[
l'(n) = L(l(n), n)
\]

with boundary condition \( l(n_0) = \bar{v}(n_0) \). Similarly, an upper bound for \( \bar{v} \) on \([n_0, \infty)\) is given by any solution \( u \) to the differential equation

\[
u'(n) = U(u(n), n)
\]

with boundary condition \( u(n_0) = \bar{v}(n_0) \). Using this analysis, we now give the full proofs of the results from Section 2.

**Proof of Theorem 1.** We begin rewriting (31) using the inequality \(|n| \leq n\) as

\[
d^+ \bar{v}(n) \geq \frac{1}{n+1} (c - (1 - \gamma)\bar{v}([n])).
\]

Then, the same inequality \(|n| \leq n\) together with Lemma 1 imply that \(\bar{v}([n]) \leq \bar{v}(n)\), which implies

\[
d^+ \bar{v}(n) \geq \frac{1}{n+1} (c - (1 - \gamma)\bar{v}(n)).
\]

Defining \( L : \mathbb{R}^2 \to \mathbb{R} \) by \( L(v, n) = \frac{1}{n+1} (c - (1 - \gamma)v) \), \( d^+ \bar{v}(n) \geq L(v(n), n) \). We solve the differential equation

\[
l'(n) = L(l(n), n) = \frac{1}{n+1} (c - (1 - \gamma)l(n))
\]

with boundary condition \( l(0) = \bar{v}(0) \).

The solution to this differential equation satisfies \( l(n) \leq \bar{v}(n) \) for all \( n \geq 0 \) and thus bounds \( \bar{v} \) from below. We solve for \( l \) using the general solution for first order linear differential equations (Boyce & DiPrima, 1997). The integrating factor is \( \mu(n) = \exp \left[ \int (1 - \gamma) \frac{dm}{m+1} \right] = \exp \left[ (1 - \gamma) \log(n+1) \right] = (n+1)^{1-\gamma} \). The solution \( l \) is given by

\[
l(n) = \frac{1}{\mu(n)} \int \mu(m) \frac{c}{m+1} \, dm
\]

\[
= c(n+1)^{-(1-\gamma)} \int (m+1)^{1-\gamma} (m+1)^{-1} \, dm
\]

\[
= c(n+1)^{-(1-\gamma)} \int (m+1)^{-\gamma} \, dm
\]

\[
= c(n+1)^{-(1-\gamma)} \left( \frac{1}{1-\gamma} (n+1)^{1-\gamma} - b \right)
\]

\[
= c \left( \frac{1}{1-\gamma} - b(n+1)^{-(1-\gamma)} \right),
\]

39
where \( b \) is an integration constant chosen so that \( l(0) = \bar{v}(0) = 0 \). We plug in \( n = 0 \) to this equation to see \( 0 = c \left( \frac{1}{1 - \gamma} - b \right) \), implying that \( b = \frac{1}{1 - \gamma} \). Thus,

\[
    l(n) = c \left( \frac{1}{1 - \gamma} - \frac{1}{1 - \gamma} (n + 1)^{-(1 - \gamma)} \right) = \frac{c}{1 - \gamma} \left( 1 - (n + 1)^{-(1 - \gamma)} \right),
\]

which completes the proof. \( \square \)

**Proof of Lemma 2.** We begin by noting, \( \bar{v}(n + 1) = \sum_{k=0}^{n} \bar{v}(k + 1) - \bar{v}(k) = \sum_{k=0}^{n} \frac{d^+}{dn} \bar{v}(k) \). By the concavity of \( \bar{v} \) as shown in Lemma 1, \( \frac{d^+}{dn} \bar{v}(k) \leq \frac{d^+}{dn} \bar{v}(n) \) for all \( k \leq n \), so \( \bar{v}(n + 1) \leq \sum_{k=0}^{n} \frac{d^+}{dn} \bar{v}(n) = (n + 1) \frac{d^+}{dn} \bar{v}(n) \). Dividing by \( n + 1 \) completes the proof. \( \square \)

**Proof of Theorem 2.** We begin by rewriting \( \bar{v}([n]) \) as

\[
    \bar{v}([n]) = \bar{v}([n] + 1) - \bar{v}([n])
    = \bar{v}([n] + 1) - \frac{d^+}{dn} \bar{v}([n])
    \geq \bar{v}([n] + 1) - \frac{\bar{v}([n] + 1)}{[n] + 1}
    \geq \bar{v}([n] + 1) - \frac{c}{[n] + 1}(1 - \gamma)
\]

where the third step is by Lemma 2, and the fourth step is by Lemma 1. We combine this with (31) to write

\[
    \frac{d^+}{dn} \bar{v}(n) \leq \frac{1}{[n] + 1} \left( c - (1 - \gamma) \bar{v}([n] + 1) + \frac{c}{[n] + 1} \right).
\]

Then, the inequality \( n \leq [n] + 1 \) implies that

\[
    \frac{d^+}{dn} \bar{v}(n) \leq \frac{1}{n} \left( c - (1 - \gamma) \bar{v}([n] + 1) + \frac{c}{n} \right).
\]

The same inequality \( n \leq [n] + 1 \) together with Lemma 1 imply that \( \bar{v}([n] + 1) \geq \bar{v}(n) \), which implies

\[
    \frac{d^+}{dn} \bar{v}(n) \leq \frac{1}{n} \left( c - (1 - \gamma) \bar{v}(n) + \frac{c}{n} \right).
\]

Defining \( U : \mathbb{R}^2 \to \mathbb{R} \) by \( U(v, n) = \frac{1}{n} \left( c - (1 - \gamma)v + \frac{c}{n} \right) \), \( \frac{d^+}{dn} \bar{v}(n) \leq U(\bar{v}(n), n) \). We solve the differential equation

\[
    u'(n) = U(u(n), n) \tag{32}
\]

with boundary condition

\[
    u(n_0) = \bar{v}(n_0). \tag{33}
\]

40
The solution, \( u \), is an upper bound for \( \bar{v} \) in the sense that \( u(n) \geq \bar{v}(n) \) for all \( n \geq n_0 \). We solve for \( u \) using the general solution for first order linear differential equations (Boyce & DiPrima, 1997).

The integrating factor \( \mu(n) \) is the integral of the term multiplying \( u(n) \),

\[
\mu(n) = \exp \left( \int (1 - \gamma)m^{-1} \, dm \right) = \exp \left( (1 - \gamma) \log(n) \right) = n^{1-\gamma}.
\]

The integral of the right hand side multiplied by the integrating factor is

\[
\int \mu(m)c(m^{-1} + m^{-2}) \, dm = c \int m^{-\gamma} + m^{-1-\gamma} \, dm = \frac{c}{1 - \gamma} \left( n^{1-\gamma} - \frac{1 - \gamma}{\gamma} n^{-\gamma} - b \right),
\]

where \( b \) is any scalar. Thus, the solution of (32) is

\[
u(n) = \frac{1}{\mu(n)} \int \mu(m)c(m^{-1} + m^{-2}) \, dm
= \frac{c}{1 - \gamma} \left( n^{1-\gamma} - \frac{1 - \gamma}{\gamma} n^{-\gamma} - b \right)
= \frac{c}{1 - \gamma} \left( 1 - \frac{1 - \gamma}{\gamma} \frac{1}{n} - bn^{-1-\gamma} \right),
\]

where \( b \) is chosen to satisfy the boundary condition (33). Solving the relation

\[
\bar{v}(n_0) = u(n_0) = \frac{c}{1 - \gamma} \left( 1 - \frac{1 - \gamma}{\gamma} \frac{1}{n_0} - bn_0^{-1-\gamma} \right)
\]

for \( b \) gives (8).

\(\Box\)

**Proof of Corollary 1.** We choose \( n_0 = 1 \) and compute \( b \). Substituting \( n_0 = 1 \) and \( \bar{v}(1) = c \) into (8) gives

\[
b = 1 - \frac{1 - \gamma}{\gamma} - \frac{1 - \gamma}{c} = -\frac{1 - \gamma}{\gamma} + \gamma = \frac{\gamma^2 + \gamma - 1}{\gamma}
\]

as required.

\(\Box\)

In Corollary 1, the constant \( b \) is strictly positive only for \( \gamma > \frac{-1 + \sqrt{5}}{2} \approx .618 \) since \( \gamma b = \gamma^2 + \gamma - 1 = \left( \gamma - \frac{1 + \sqrt{5}}{2} \right) \left( \gamma - \frac{1 - \sqrt{5}}{2} \right) \). When \( b \) is strictly positive, Corollary 1 provides a useful bound on the asymptotic convergence of \( \bar{v} \). When \( b \) is negative, however, as \( n \) becomes large the upper bound \( \frac{c}{1 - \gamma} \left[ 1 - bn^{-1-\gamma} - \frac{1 - \gamma}{\gamma} \frac{1}{n} \right] \) becomes larger than the trivial upper bound \( \frac{c}{1 - \gamma} \) shown in Lemma 1 and the bound is no longer useful. To obtain useful bounds from Theorem 2 for a broader range of \( \gamma \) we may increase the \( n_0 \) chosen. Increasing \( n_0 \) also increases \( b \) and tightens the bound across all \( \gamma \).
Appendix: The role of $1/n$ in approximate policy iteration

The recursive least-squares (RLS) method uses a value function approximation of the form

$$\bar{V}(S_t) = \bar{\theta}_0 + \sum_{i=1}^{I} \bar{\theta}_i \phi_i(S_t)$$

where $\phi_1, ..., \phi_I$ are basis functions mapping a state to a vector in Euclidean space. In RLS, we construct $\bar{V}^n$ by solving the least-squares problem

$$\min_{\theta} \sum_{m=1}^{n} \left( \hat{v}^m - \left( \theta_0 + \sum_{i=1}^{I} \theta_i \phi_i(S^m) \right) \right)^2$$

and letting $\bar{\theta}^n$ be the solution. The approximation can be updated recursively using the equation (see Powell, 2007, sec. 7.3.2)

$$\bar{\theta}^n = \bar{\theta}^{n-1} - \frac{1}{1 + (x^n)^T B^{n-1} x^n} B^{n-1} x^n (\bar{V}^{n-1} (S^n) - \hat{v}^n)$$

where $x^n = (1, \phi_1(S^n), ..., \phi_I(S^n))$, and the matrix $B^n$ is also found recursively using the equation

$$B^n = B^{n-1} - \frac{1}{1 + (x^n)^T B^{n-1} x^n} \left( B^{n-1} x^n (x^n)^T B^{n-1} \right).$$

There is no explicit stepsize in these calculations. However, we can see RLS in a new light if we consider the single-state, single-action problem. In this simple MDP, there is only one basis function $\phi \equiv 1$ and one parameter $\bar{\theta} = \bar{V}$. The matrix $B^n$ becomes a scalar, and (35) reduces to

$$\bar{v}^n = \bar{v}^{n-1} - \frac{B^{n-1}}{1 + B^{n-1}} (\bar{v}^{n-1} - \hat{v}^n) = \left( 1 - \frac{B^{n-1}}{1 + B^{n-1}} \right) \bar{v}^{n-1} + \frac{B^{n-1}}{1 + B^{n-1}} \hat{v}^n.$$ 

Furthermore, $B^n = \frac{B^{n-1}}{1 + B^{n-1}}$. If $B^0 = 1$, then $B^{n-1} = 1/n$, and the RLS equations reduce to (5) with the $1/n$ stepsize rule. We see that sample averaging is implicitly present in the RLS method, which at first glance would seem to avoid using any stepsize rule. This insight serves to underscore the importance of our results on the slow convergence of the $1/n$ rule.

We can also carry this discussion over to the setting of approximate policy iteration, where it provides additional insight into the way such algorithms learn, as well as the role played by sample averaging. Recall that least-squares methods approximate the value function as a linear combination

$$\bar{V}(S_t) = \bar{\theta}_0 + \sum_{i=1}^{I} \bar{\theta}_i \phi_i(S_t)$$
where $\phi_1, \ldots, \phi_I$ are basis functions mapping a state to a vector in Euclidean space. The LSPE procedure of Nediç & Bertsekas (2003) computes the value of a policy by generating a trajectory $(S^0, S^1, \ldots)$ of states visited by that policy, and computing (Bertsekas, 2007, eq. 6.31)

$$\bar{\theta}^n = \arg \min_{\theta} \sum_{m=1}^{n} \left( \theta_0 + \sum_{i=1}^{I} \theta_i \phi_i(S^m) - \hat{c}^m - \gamma \bar{V}^{n-1}(S^{m+1}) \right)^2.$$ 

For the single-state, single-action MDP, there is a closed-form solution

$$\bar{v}^n = \arg \min_{\theta} \frac{1}{n} \sum_{m=1}^{n} \left( \hat{c}^m + \gamma \bar{v}^{n-1} - \theta \right)^2 = \left( \frac{1}{n} \sum_{m=1}^{n} \hat{c}^m \right) + \gamma \bar{v}^{n-1}.$$

In this problem, the LSPE procedure is equivalent to the value iteration equation $v^n = c + \gamma v^{n-1}$, with $c$ replaced by the sample average. Essentially, LSPE is running value iteration under the assumption that the mean one-period reward is unknown. LSPE is implicitly using a $1/n$ stepsize rule to estimate the single-period cost, and is then using a pure summation to add up the value. In a general MDP, the observations of $\hat{c}^n$ will generally be non-stationary, but a $1/n$ stepsize rule may work reasonably well.

The LSTD procedure of Bradtke & Barto (1996) computes $\bar{\theta}^n$ by solving the system of linear equations (see Bertsekas, 2007, eq. 6.54) given by

$$\sum_{m=1}^{n} \phi_i(S^m) \left( \phi_i(S^m) - \gamma \phi_i(S^{m+1}) \right)^T \theta = \sum_{m=1}^{n} \phi(S^m) \hat{c}^m, \quad i = 0, \ldots, I$$

for $\theta$. For the single-state, single-action MDP, the solution is given by

$$\bar{v}^n = \frac{1}{1 - \gamma} \frac{1}{n} \sum_{m=1}^{n} \hat{c}^m,$$

the infinite-horizon discounted sample average of the one-period rewards obtained thus far. Instead of successively summing values, LSTD estimates the average contribution per period and then projects the infinite-horizon value of the policy.

We see that both LSPE and LSTD involve the computation of a sample average, which is equivalent to smoothing an estimate with a stepsize of $1/n$. However, these methods have the advantage of being able to isolate the stepsize issue to the estimation of a stationary quantity, namely the mean $c$ of the one-period reward. For a general dynamic program, however, the observations $\hat{c}^n$ will be non-stationary because the probability of being in a state (as well as the probability of choosing an action) will be evolving in the early iterations. In larger problems with non-stationary data, especially problems where there is no obvious choice of basis functions or where it takes a long time
to reach steady state, approximate value iteration is often the most intuitive method, and requires
the least computational effort due to the ability to bootstrap from an existing approximation in
each time step.