TD(0) Converges Provably Faster than the Residual Gradient Algorithm

Ralf Schoknecht ralf.schoknecht@ilkd.uni-karlsruhe.de
Institute of Logic, Complexity and Deduction Systems, University of Karlsruhe, 76128 Karlsruhe, Germany

Artur Merke artur.merke@udo.edu
Lehrstuhl Informatik 1, University of Dortmund, 44227 Dortmund, Germany

Abstract

In Reinforcement Learning (RL) there has been some experimental evidence that the residual gradient algorithm converges slower than the TD(0) algorithm. In this paper, we use the concept of asymptotic convergence rate to prove that under certain conditions the synchronous off-policy TD(0) algorithm converges faster than the synchronous off-policy residual gradient algorithm if the value function is represented in tabular form. This is the first theoretical result comparing the convergence behaviour of two RL algorithms. We also show that as soon as linear function approximation is involved no general statement concerning the superiority of one of the algorithms can be made.

1. Introduction

Reinforcement Learning (RL) is concerned with finding optimal policies for dynamical optimisation problems. An agent interacting with its environment selects actions depending on the current state of the environment. As a result it obtains a reward and the environment makes a transition to a new state. The objective of the agent is to optimise its policy, i.e. the mapping from states to actions, through learning. For a fixed policy the utility of a state is given by the expected accumulated discounted reward that is received over time when starting in that state and behaving according to the fixed policy. One method to learn an optimal policy is policy iteration (Bertsekas & Tsitsiklis, 1996). This algorithm consists of two steps that are executed in turns. In the step called policy evaluation the agent learns a value function that represents these utilities for every state and a fixed policy. Following that the policy is changed in the policy improvement step and the new policy is evaluated again. For the problem of policy evaluation a popular algorithm is the TD(0) algorithm (Sutton, 1988). This algorithm is guaranteed to converge if the value function is represented by a table, i.e. every state has a separate entry to store the value. For large problems, however, such a tabular representation is no longer feasible with respect to time and memory considerations. Therefore, linear feature-based function approximation is often used. However, in the presence of linear function approximation the TD(0) algorithm may diverge when transitions are arbitrarily sampled. For that reason the residual gradient (RG) algorithm has been proposed as a convergent alternative (Baird, 1995).

Baird (1995) also stated the phenomenon of slow convergence of the RG algorithm by giving an example. However, the reasons for slow convergence were not investigated further. In this paper we use the asymptotic convergence rate to measure the speed of convergence. We formulate the TD(0) and the RG algorithm as linear iterations. Our convergence analysis is based on the eigenvalues of the iteration matrices corresponding to the two algorithms. Using such spectral methods in RL proves quite powerful. We show that under certain conditions the TD(0) algorithm converges asymptotically faster than the RG algorithm if a tabular representation is used for the value function. This is the first theoretical result comparing the convergence behaviour of two RL algorithms. We apply our theorem to the rooms gridworld benchmark of Sutton et al. (1999). For a stochastic random walk policy and for an optimal deterministic policy we show that the TD(0) algorithm converges asymptotically faster than the RG algorithm as predicted by our theorem. We also demonstrate that in the example of Baird with function approximation the RG algorithm may converge asymptotically faster than the TD(0) algorithm. Thus, the TD(0) algorithm is not always superior.

2. Synchronous Reinforcement Learning

Before we investigate the convergence speed of the TD(0) and the RG algorithm we first formally introduce these two algorithms in their synchronous off-policy versions.

2.1. Approximate Policy Evaluation

For a Markov decision process (MDP) with finite state space $S = \{s_1, \dots, s_N\}$, finite action space A, state transition probabilities $p : (S, S, A) \rightarrow [0,1]$ and stochastic reward function $r : (S, A) \rightarrow \mathbb{R}$ policy evaluation is concerned with solving the Bellman equation

$$V^\pi = \gamma P^\pi V^\pi + R^\pi \tag{1}$$

for a fixed policy $\pi : S \rightarrow A$. V_i^π denotes the value of state s_i, $P_{i,j}^\pi = p(s_i, s_j, \pi(s_i))$, $R_i^\pi = E\{r(s_i, \pi(s_i))\}$ and γ is the discount factor. As the policy π is fixed we will omit it in the following to make notation easier.

If the state space S gets too large the exact solution of equation (1) becomes very costly with respect to both memory and computation time. Therefore, often linear feature-based function approximation is applied. The value function V is represented as a linear combination of F basis functions $H := \{\Phi_1, \dots, \Phi_F\}$ which can be written as $V = \Phi w$, where $w \in \mathbb{R}^F$ is the parameter vector describing the linear combination and $\Phi = (\Phi_1 | \dots | \Phi_F) \in \mathbb{R}^{N \times F}$ is the matrix with the basis functions as columns. The rows of Φ are the feature vectors $\varphi(s_i) \in \mathbb{R}^F$ for the states s_i.

The approximate policy evaluation problem derived from (1) is then given by $\Phi w = \gamma P \Phi w + R$. If this problem were exactly solvable we obtained

$$(\gamma P - I_N)\Phi w + R = 0, \tag{2}$$

where I_N is the identity in $\mathbb{R}^{N \times N}$. However, in general there will be no w such that this equation is fulfilled. Therefore, we have to trade off the errors made in the different components. This is done by a diagonal weighting matrix D which yields the quadratic error measure based on the norm[1] $\|\cdot\|_D$ that is known as Bellman error

$$\begin{aligned} E_B(w) &= \frac{1}{2}[(\gamma P - I_N)\Phi w + R]^\top D[(\gamma P - I_N)\Phi w + R] \\ &= \frac{1}{2}\|(\gamma P - I_N)\Phi w + R\|_D^2. \end{aligned} \tag{3}$$

[1]For $x \in \mathbb{R}^N$ a symmetric and positive definite matrix $B \in \mathbb{R}^{N \times N}$ defines a norm according to $\|x\|_B = \sqrt{x^\top Bx}$.

2.2. The TD(0) Algorithm

The TD(0) algorithm (Sutton, 1988) is one way to solve the approximate policy evaluation problem. It is generally stated as an asynchronous algorithm, that updates the parameters after each transition. We do not require the transitions to be sampled in an on-policy manner along a trajectory corresponding to the policy that is evaluated. They can rather be sampled off-policy. Let us assume that a transition $x_i \rightarrow z_i$ with reward r_i is observed. The TD(0) learning rule corresponding to this single transition updates the parameters w of a general linear function approximator. With the learning rate α this TD(0) update can be written in vector notation as follows

$$\begin{aligned} w^{n+1} &= w^n + \alpha\varphi(x_i)[r_i + \gamma V(z_i) - V(x_i)] \\ &= w^n + \alpha\varphi(x_i)[r_i + \gamma\varphi^\top(z_i)w^n - \varphi^\top(x_i)w^n] \\ &= (I_F + \alpha A_i)w^n + \alpha b_i, \end{aligned}$$

where $A_i = \varphi(x_i)[\gamma\varphi(z_i) - \varphi(x_i)]^\top$, $b_i = \varphi(x_i)r_i$ and I_F denotes the identity in $\mathbb{R}^{F \times F}$.

Often a whole set of transitions has been sampled from the system that is to be controlled. If these transitions are stored in a database they can also be used for a synchronous update. Such synchronous update rules are more convenient for a mathematical convergence analysis than the corresponding asynchronous versions. There is evidence that a synchronous update rule approximates the behaviour of asynchronous update rules in the limit of the iteration (Bertsekas & Tsitsiklis, 1996). Thus, the study of synchronous update rules will also be useful to better understand the behaviour of asynchronous update rules. The situation is similar to the relation of batch learning and pattern learning in supervised learning. From now on, by referring to the TD(0) and the RG algorithm we mean the synchronous off-policy versions of these two algorithms.

In the following we consider the synchronous update for a fixed set of m arbitrary transitions denoted by $T = \{(x_i, z_i, r_i) | i = 1, \dots, m\}$. The start states x_i are sampled with respect to the probability distribution ρ, the next states z_i are sampled according to $p(x_i, \cdot)$ and the rewards r_i are sampled from $r(x_i)$. The synchronous update for the transition set T can then be written in matrix notation as

$$w^{n+1} = (I_F + \alpha A_{TD})w^n + \alpha b_{TD} \tag{4}$$

with $A_{TD} = A_1 + \dots + A_m$ and $b_{TD} = b_1 + \dots + b_m$. Let $X \in \mathbb{R}^{m \times N}$ with $X_{i,j} = 1$ if $x_i = s_j$ and 0 otherwise. Then, $X\Phi \in \mathbb{R}^{m \times F}$ is the matrix with feature vector $\varphi(x_i)$ as its i-th row. We define $Z \in \mathbb{R}^{m \times N}$ accordingly with $Z_{i,j} = 1$ if $z_i = s_j$ and 0 otherwise. Then, the matrix $Z\Phi \in \mathbb{R}^{m\times F}$ contains the feature vector $\varphi(z_i)$ as its i-th row. With the vector of rewards $r = (r_1, \dots, r_m)^\top$ this yields

$$A_{TD} = \Phi^\top X^\top(\gamma Z\Phi - X\Phi), \quad b_{TD} = \Phi^\top X^\top r. \tag{5}$$

2.3. A Different Representation

We have stated the TD(0) algorithm in terms of X, Z and r. In the following we will show how these entities relate to D, P and R that are used in (3). We define the matrix $\widehat{D} = X^\top X$ which is diagonal and contains the absolute frequencies with which the states s_i occur as starting states in the transition set T. The entry (i,j) of matrix $X^\top Z$ can be interpreted as the absolute frequency of transition $s_i \to s_j$. We define $\widehat{P}$ as the matrix containing the respective relative frequencies, i.e. $\widehat{P} = (1/\widehat{D}_{i,i}) \cdot (X^\top Z)_{i,j}$ if $\widehat{D}_{i,i} \neq 0$ and $\widehat{P} = 0$ otherwise. This yields $\widehat{D}\widehat{P} = X^\top Z$. Furthermore, the i-th entry of the vector $X^\top r$ contains the sum of rewards that have been obtained on transitions from state s_i. With the vector $\widehat{R}$ containing the average rewards that occur on transitions from the different starting states s_i we obtain $\widehat{D}\widehat{R} = X^\top r$.

Thus, the entities $\widehat{P}$, $\widehat{R}$ and $\widehat{D}$ can be computed from the sampled transition data T. It can be shown that as the number of transitions m goes to infinity the entities $\widehat{P}$, $\widehat{R}$ and $\frac{1}{m}\widehat{D}$ converge to P, R and D with probability one. In the case of a finite sample size we have estimates of the true entities and the Bellman error (3) becomes

$$E_B(w) = \frac{1}{2m}\|(\gamma\widehat{P} - I_N)\Phi w + \widehat{R}\|^2_{\widehat{D}}. \tag{6}$$

Moreover, we have deduced a relation between X, Z, r and $\widehat{P}$, $\widehat{R}$, $\widehat{D}$. The TD(0) algorithm given by (5) can therefore be represented in an equivalent form, which will be important for the proof of our main theorem

$$A_{TD} = \Phi^\top \widehat{D}(\gamma\widehat{P} - I_N)\Phi, \quad b_{TD} = \Phi^\top \widehat{D}\widehat{R}. \tag{7}$$

2.4. The Residual Gradient Algorithm

In case function approximation is used the residual gradient (RG) algorithm was introduced as a convergent alternative to the possibly divergent TD(0) algorithm. The RG algorithm directly minimises the Bellman error by gradient descent. The gradient of (6) is given by

$$\frac{\partial E_B(w)}{\partial w} = \frac{1}{m}[(\gamma\widehat{P} - I_N)\Phi]^\top \widehat{D}[(\gamma\widehat{P} - I_N)\Phi w + \widehat{R}].$$

From this gradient we can deduce the update rule of the RG algorithm. It has a form similar to (4)

$$w^{n+1} = (I_F + \alpha A_{RG})w^n + \alpha b_{RG} \tag{8}$$

with $A_{RG}w + b_{RG} = -m\frac{\partial E_B(w)}{\partial w}$. This update rule uses the negative gradient and moves the factor $\frac{1}{m}$ into the learning rate α. For A_{RG} and b_{RG} we obtain

$$\begin{aligned} A_{RG} &= -\Phi^\top[(\gamma\widehat{P} - I_N)]^\top \widehat{D}[(\gamma\widehat{P} - I_N)]\Phi, \\ b_{RG} &= -\Phi^\top[(\gamma\widehat{P} - I_N)]^\top \widehat{D}\widehat{R}. \end{aligned} \tag{9}$$

This representation will be used in the proof of the main theorem, which we will state in the next section.

3. Asymptotic Convergence

3.1. Asymptotic Convergence Rate

The iteration of the TD(0) and the RG algorithm is of the form

$$w^{n+1} = (I_F + \alpha A)w^n + \alpha b. \tag{10}$$

In the following we investigate the asymptotic convergence properties of such iterations. Let w^* be the fixed point of iteration (10), i.e. $Aw^* + b = 0$ holds. And let $e^n = (w^n - w^*)$ be the error after n steps. This error evolves according to the following iteration

$$\begin{aligned} e^{n+1} &= w^{n+1} - w^* \\ &= (I_F + \alpha A)w^n + \alpha b - w^* - \alpha\underbrace{(Aw^* + b)}_{=0} \\ &= (I_F + \alpha A)(w^n - w^*) = (I_F + \alpha A)e^n. \end{aligned}$$

From this iterative formula we obtain the following explicit representation

$$e^n = (I_F + \alpha A)^n e^0. \tag{11}$$

Instead of the error vector e^n we consider the norm $\|e^n\|$ of this vector, where $\|\cdot\|$ denotes an arbitrary vector norm in $\mathbb{R}^F$. Every vector norm in $\mathbb{R}^F$ induces a compatible matrix norm in $\mathbb{R}^{F\times F}$, which we also denote by $\|\cdot\|$. It is defined as follows

$$\|A\| := \max_{x\in\mathbb{R}^F\setminus\{0\}} \frac{\|Ax\|}{\|x\|} = \max_{x\in\mathbb{R}^F, \|x\|=1} \|Ax\|. \tag{12}$$

Asymptotic convergence is concerned with the question, how the error $\|e^n\|$ after n steps is related to the initial error $\|e^0\|$ in the worst case if n goes to infinity. This leads to the notion of asymptotic convergence rate, which is defined in Greenbaum (1997) and Varga (1962) for nonsingular matrices A.

Definition 1 *Let e^n be the n-th error vector of iteration* (10)*. Then*

$$c := \lim_{n\to\infty} \max_{e^0\in\mathbb{R}^F\setminus\{0\}} (\frac{\|e^n\|}{\|e^0\|})^{\frac{1}{n}} \tag{13}$$

is the asymptotic convergence rate *of this iteration.*

Thus, the asymptotic convergence rate indicates by what factor the error e^n is at least reduced asymptotically in every step on the average.

In the following we will see that the spectral radius of the iteration matrix plays a crucial role in the determination of the asymptotic convergence rate. The spectral radius of a matrix M is the absolute value of the eigenvalue of M with largest absolute value, i.e. $\varrho(M) = \max_{\lambda \in \sigma(M)} |\lambda|$, where $\sigma(M)$ is the set of all eigenvalues of M. According to Corollary 1.3.1 in Greenbaum (1997) for every matrix norm $\|\cdot\|$ and every Matrix M with spectral radius $\varrho(M)$ the following holds

$$\lim_{n\to\infty} \|M^n\|^{\frac{1}{n}} = \varrho(M). \tag{14}$$

For the asymptotic convergence rate we therefore obtain

$$\begin{aligned} c &\overset{(13)}{=} \lim_{n\to\infty} \max_{e^0 \in \mathbb{R}^F \setminus \{0\}} \left(\frac{\|e^n\|}{\|e^0\|}\right)^{\frac{1}{n}} \\ &\overset{(11)}{=} \lim_{n\to\infty} \max_{e^0 \in \mathbb{R}^F \setminus \{0\}} \left(\frac{\|(I+\alpha A)^n e^0\|}{\|e^0\|}\right)^{\frac{1}{n}} \\ &\overset{(12)}{=} \lim_{n\to\infty} (\|(I+\alpha A)^n\|)^{\frac{1}{n}} \overset{(14)}{=} \varrho(I+\alpha A). \end{aligned} \tag{15}$$

3.2. Optimal Learning Rate

This asymptotic convergence rate is sharp for nonsingular matrices because there is at least one error vector that decreases according to this rate, namely the eigenvector of $I_F + \alpha A$ corresponding to the spectral radius $\varrho(I+\alpha A)$. Therefore, the iteration (10) is asymptotically governed by $(\varrho(I+\alpha A))^n$. This entity depends on the learning rate α. Hence, we can deduce an optimisation criterion for the learning rate.

Lemma 1 *Let A be a nonsingular matrix. If there exists a learning rate such that iteration* (10) *converges then the optimal asymptotic convergence rate is obtained for the learning rate*

$$\alpha^* := \arg\min_{\alpha} \max_{\lambda \in \sigma(A)} |1+\alpha\lambda|.$$

We denote α^* as optimal learning rate*.*

Proof: According to (15) the asymptotic convergence rate depends on the learning rate α. From the definition of the spectral radius it follows that $c(\alpha) = \max_{\lambda\in\sigma(A)} |1+\alpha\lambda|$. Thus, the optimal asymptotic convergence rate is obtained by minimising over α which yields $c^* = \min_\alpha c(\alpha)$. And therefore the optimal learning rate is $\alpha^* = \arg\min_\alpha c(\alpha)$. □

In case A has only real eigenvalues the following corollary yields a particularly simple expression for α^*.

Corollary 1 *Let A be a nonsingular matrix that has only negative real eigenvalues $\{\lambda_1, \dots, \lambda_n\} \subset \mathbb{R}^{<0}$ ordered such that $\lambda_1 \le \ldots \le \lambda_F < 0$. Then the optimal learning rate of iteration* (10) *is given by*

$$\alpha^* := \frac{2}{|\lambda_1| + |\lambda_F|}.$$

Proof: The eigenvalues of $I+\alpha A$ lie in the interval $[1+\alpha\lambda_1, 1+\alpha\lambda_F]$. Thus, $\varrho(I+\alpha A)$ is either $|1+\alpha\lambda_1|$ or $|1+\alpha\lambda_F|$. It can easily be seen that ϱ is minimal if the two eigenvalues lie symmetrically around zero. Therefore, $1+\alpha^*\lambda_1 < 0$, $1+\alpha^*\lambda_F > 0$ and

$$\begin{aligned} &|1+\alpha^*\lambda_1| = |1+\alpha^*\lambda_F| \iff -(1+\alpha^*\lambda_1) = 1+\alpha^*\lambda_F \\ &\iff \alpha^*(-\lambda_1 - \lambda_F) = 2 \iff \alpha^* = \frac{2}{|\lambda_1|+|\lambda_F|}. \end{aligned}$$

□

Note, the optimal learning rate is usually not the largest possible learning rate that leads to convergence of the iteration (10). Therefore, a larger learning rate does not necessarily lead to faster asymptotic convergence (Schoknecht & Merke, 2003).

3.3. The Bellman Error Measure

In section 3.1, we used $\|e^n\| = \|w^n - w^*\|$ to measure the convergence speed of RL algorithms, where $\|\cdot\|$ is an arbitrary vector norm. However, in reinforcement learning the Bellman error is generally used instead. In the following we will show that this is no contradiction for a quasi tabular representation, i.e. with nonsingular $\Phi \in \mathbb{R}^{N\times N}$. In this case, the Bellman error can be expressed as $\|e^n\|^2$ with respect to a certain norm for both the TD(0) and the RG algorithm. From (2) we can deduce that $\widehat{R} = -(\gamma\widehat{P} - I_N)\Phi w^*$ holds for the approximate entities $\widehat{P}$ and $\widehat{R}$. Thus, for the Bellman error (6) we obtain

$$\begin{aligned} E_B(w^n) &= \frac{1}{2m}\|(\gamma\widehat{P} - I_N)\Phi w^n + \widehat{R}\|^2_{\widehat{D}} \\ &= \frac{1}{2m}\|(\gamma\widehat{P} - I_N)\Phi w^n - (\gamma\widehat{P} - I_N)\Phi w^*\|^2_{\widehat{D}} \\ &= \frac{1}{2m}\|w^n - w^*\|^2_{\Phi^\top(\gamma\widehat{P}-I_N)^\top\widehat{D}(\gamma\widehat{P}-I_N)\Phi}. \end{aligned}$$

As Φ was assumed to be nonsingular the matrix $B = \Phi^\top(\gamma\widehat{P}-I_N)^\top\widehat{D}(\gamma\widehat{P}-I_N)\Phi \in \mathbb{R}^{N\times N}$ is symmetric and positive definite and therefore defines a norm $\|\cdot\|_B$. Thus, the Bellman error is equivalent to the norm of e^n with respect to this matrix.

However, if function approximation is used, i.e. Φ is singular, the TD(0) algorithm and the RG algorithm in general converge to different solutions w^*_{TD} and

w^*_{RG} respectively. In this case the TD(0) algorithm no longer minimises the Bellman error and $\|w^n - w^*_{TD}\|^2$ has to be considered instead (Schoknecht, 2003).

3.4. Comparison of TD(0) and RG

For a certain class of problems the following theorem states that the synchronous off-policy TD(0) algorithm converges asymptotically faster than the synchronous off-policy RG algorithm. The proof can be found in the appendix

Theorem 1 *Let $(I_F + \alpha_{TD}A_{TD})$ and $(I_F + \alpha_{RG}A_{RG})$ be the iteration matrices of the synchronous TD(0) and the synchronous RG algorithm respectively. We assume that the corresponding value functions are represented in tabular form ($\Phi = I_N$, $F = N$) and that the set of transitions contains each starting state with equal frequency ($\widehat{D} = kI_N, k \in \mathbb{N}$). Assume that the matrix A_{TD} has only real eigenvalues (which is always the case for A_{RG}). Let the learning rates α^*_{TD} and α^*_{RG} be the optimal learning rates according to Corollary 1. Moreover, let not all of the eigenvalues of A_{RG} be identical. Then, the TD(0) algorithm converges asymptotically faster than the RG algorithm. If all eigenvalues of A_{RG} are identical, then both the RG algorithm and the TD(0) algorithm converge equally fast asymptotically.*

This is the first theoretical result that compares the speed of convergence of the TD(0) and the RG algorithm. Baird (1995) only stated the phenomenon of slow convergence of the RG algorithm by giving an example. However, the reasons for slow convergence were not investigated further. We use the asymptotic convergence rate to prove that the TD(0) algorithm converges asymptotically faster than the RG algorithm. Our result is restricted to the class of matrices A_{TD} with real eigenvalues that represent a uniform sampling of the state transitions and a tabular representation of the value function.

Although the RG algorithm has originally been formulated as an alternative to the TD(0) algorithm to ensure convergence for an approximate representation of the value function it can also be used with a tabular representation. However, for a tabular representation the TD(0) algorithm is known to converge and the RG algorithm would not be necessary. Nevertheless, we think the problem of a tabular representation is worth studying because it constitutes the basic problem. Understanding the mechanisms that are important for convergence in this basic problem can serve as a basis to study more general problems with function approximation. Even if no general result concerning the convergence speed of RL algorithms with function approximation is available yet, the concept of asymptotic convergence rate can be used to analyse individual problems with function approximation. In section 5 we will fully analyse the convergence speed of the two RL algorithms applied to the example of Baird (1995). In this example the eigenvalues of A_{TD} can also be complex (Schoknecht & Merke, 2003). This shows that our analysis is not only applicable if the eigenvalues are real. However, theoretical results for complex eigenvalues do not exist yet.

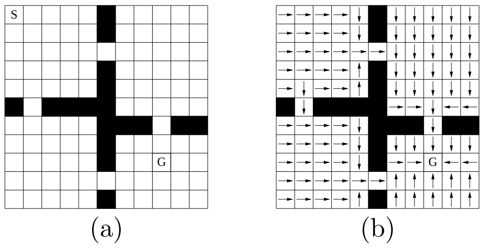

Figure 1. (a) Rooms gridworld. (b) Rooms gridworld with an optimal deterministic policy

The above theorem involves the optimal learning rate that is only computable if the eigenvalues of the matrix A are known. In practice this is not always the case. However, the theorem indicates the behaviour in the optimal case and, as the asymptotic convergence rate is a continuous function of the learning rate, we can also expect the TD(0) algorithm to converge faster than the RG algorithm in the neighbourhood of the respective optimal learning rates.

4. Applications

In this section we apply Theorem 1 to the rooms gridworld (Sutton et al., 1999), a benchmark for reinforcement learning. We demonstrate that the TD(0) algorithm converges faster than the RG algorithm as predicted by Theorem 1. However, it will also become clear, that faster convergence should always be seen in an asymptotical sense.

Figure 1(a) shows the rooms gridworld. It consists of four rooms each of them having 2 doors. The example has 121 states with one goal state G. The position of the goal state determines the reward structure: every transition into the goal state has reward 1, all other transitions have reward zero.

We will consider two different transition sets T_r and T_d. T_d belongs to an optimal deterministic policy, and T_r corresponds to a random walk policy. We begin with T_r that contains four transitions for every state corresponding to the four actions {up,down,right,left}. If a transition results in a crash with one of the walls, the transition is a self transition, otherwise it is a reg-

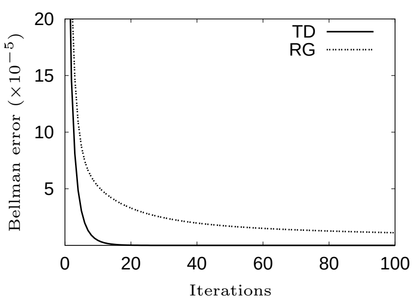

Figure 2. Random walk Bellman error.

ular transition to the neighbouring state. We use as discount factor $\gamma = 0.9$. The matrix $A_{TD} \in \mathbb{R}^{121\times 121}$ computed according to (7) has only real eigenvalues in the range $[-7.0482, -0.4]$, which, according to Corollary 1, results in the optimal learning rate $\alpha^*_{TD} \approx 0.2685$. The matrix A_{RG} computed according to (9) has also only real eigenvalues in the range $[-12.4194, -0.0151]$ and therefore $\alpha^*_{RG} \approx 0.1608$. In Figure 2 we can see that the TD algorithm quickly reaches a very small Bellman error. After 8 iterations the error drops below 10^{-5}, whereas the RG algorithms decays quite slowly reaching a Bellman error of approximately 10^{-5} only after more than 100 iterations. This strong discrepancy can be explained by comparing the spectral radii

$$\varrho(I_F+\alpha^*_{TD}A_{TD}) \approx 0.893 < 0.996 \approx \varrho(I_F+\alpha^*_{RG}A_{RG})$$

where the spectral radius of the RG algorithm is very close to one. According to the argument in section 3.2 the learning curves are asymptotically similar to 0.893^n and 0.996^n, which explains the faster convergence of the TD(0) algorithm.

We now consider our second transition set T_d. Here, from every state we have a transition with respect to an optimal policy. There are many optimal policies so we depicted the one used here in Figure 1(b). We again choose $\gamma = 0.9$ as the discount factor. The matrix A_{TD} has only two eigenvalues, namely -1 and -0.1 which results in the optimal learning rate $\alpha^*_{TD} \approx 1.818$. The matrix A_{RG} has only real eigenvalues in the range $[-5.3231, -0.00033]$ and therefore, according to Corollary 1, $\alpha^*_{RG} \approx 0.376$. For the spectral radii we obtain

$$\varrho(I_F+\alpha^*_{TD}A_{TD}) \approx 0.818 < 0.999 \approx \varrho(I_F+\alpha^*_{RG}A_{RG})$$

where now the difference is even stronger then in the random walk case. In Table 1 some values of the TD(0) iteration and the RG iteration are depicted. In step 347 the TD(0) algorithm for the first time has a smaller Bellman error than the RG algorithm and the error is decreasing much more rapidly. However, in the beginning of the iteration the error of the TD(0) algorithm rises extremely, whereas the error of the RD

Table 1. Development of the Bellman error for the deterministic gridworld problem.

Iteration	TD	RG
0	0.0165289	0.0165289
1	0.0995834	0.00966669
2	0.511085	0.00864146
8	12494.7	0.00829222
83	2.32207e+24	0.00811757
294	80481.7	0.00769193
333	0.58628	0.00761576
339	0.0911655	0.0076041
347	0.00750337	0.00758859
348	0.00550166	0.00758666
349	0.00400741	0.00758472
350	0.00293998	0.00758279
382	2.05426e-07	0.00752111

algorithm slowly but constantly falls due to its gradient descent property. The reason for the behaviour of the TD(0) algorithm can be explained by looking at the iteration in the generalized eigenspaces. We will not discuss this in detail here but just give a very brief idea of what is happening. The convergence rate of the iteration is determined by how fast the powers of the Jordan normal form of $I_F + \alpha^* A$ approach diagonal form. These powers contain entries of the form $(J^k)_{i,j} = \binom{k}{i-j}(1+\lambda)^{k-(i-j)}$ (Greenbaum, 1997), where λ is an eigenvalue of A and $i-j$ denotes the distance from the main diagonal. In high dimensional generalized eigenspaces $i-j$ can be large. As a consequence, in the beginning the binomial coefficients dominate the matrix J^k and it takes longer for the powers $(1+\lambda)^{k-(i-j)}$ to annihilate the binomial coefficients. This is what happens when the TD(0) algorithm is applied to the deterministic example because the largest generalised eigenspace of $I_F + \alpha^*_{TD}A_{TD}$ has dimension 16. However, after the initial rise the iteration values drop quite quickly so that the iteration converges asymptotically faster than for the RG algorithm.

5. Reinforcement Learning with Linear Function Approximation

As soon as linear function approximation is involved the TD(0) algorithm need not converge faster than the RG algorithm. We show this using the example of Baird (Baird, 1995) with $F = 8$ features, $N = 7$ states and $m = 7$ transitions. In this example the matrices Φ and $\widehat{P}$ are given by

$$\Phi = \begin{pmatrix} 1\,2\,0\,0\,0\,0\,0\,0 \\ 1\,0\,2\,0\,0\,0\,0\,0 \\ 1\,0\,0\,2\,0\,0\,0\,0 \\ 1\,0\,0\,0\,2\,0\,0\,0 \\ 1\,0\,0\,0\,0\,2\,0\,0 \\ 1\,0\,0\,0\,0\,0\,2\,0 \\ 2\,0\,0\,0\,0\,0\,0\,1 \end{pmatrix}, \quad \widehat{P} = \begin{pmatrix} 0\,0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,1 \end{pmatrix}$$

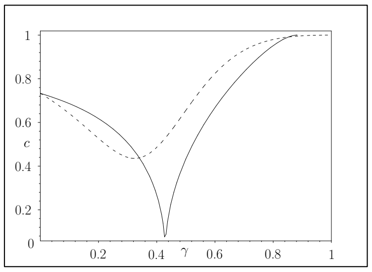

Figure 3. Asymptotic convergence rate c for the TD(0) algorithm (solid) and the RG algorithm (dotted) depending on the discount factor γ. The learning rate is set to the optimal value for both algorithms respectively.

The matrix $\widehat{D}$ equals the identity matrix I_N in $\mathbb{R}^{N\times N}$ and the vector of rewards is given by $\widehat{R} = 0$. With (7) and (9) we obtain the iterations (4) and (8) of the TD(0) and the RG algorithm respectively. Both the matrix A_{TD} and A_{RG} are singular. It can be shown that in this case the asymptotic convergence rate is given by the largest eigenvalue unequal one if the iteration converges. According to Lemma 1 we choose the optimal learning rates α^*_{TD} and α^*_{RG} for each of the two algorithms in this example. Figure 5 shows the asymptotic convergence rate c for the two algorithms depending on the discount factor γ, where a smaller value of c yields faster convergence. This shows that the TD(0) algorithms converges asymptotically faster than the RG algorithm only for $\gamma \in (0.34, 0.85)$. For all other γ the RG algorithm converges asymptotically faster than the TD(0) algorithm. Note, that the TD(0) algorithm even diverges for $\gamma \in (0.88, 1)$ (Schoknecht & Merke, 2003).

6. Conclusions

Until now there was only experimental evidence that in many cases the residual gradient (RG) algorithm converges slower than the TD(0) algorithm. In this paper, we stated the first theoretical result comparing the asymptotic convergence behaviour of the two algorithms. We proved that under certain conditions the TD(0) algorithm converges asymptotically faster than the RG algorithm if a tabular representation is used for the value function.

We applied our theorem to the rooms gridworld benchmark of Sutton et al. (1999). For a stochastic random walk policy and for an optimal deterministic policy we showed that the TD(0) algorithm converges asymptotically faster than the RG algorithm as predicted by our theorem. For the example of Baird (1995) that involves linear function approximation we demonstrated that each of the two algorithms can converge faster than the other depending on the discount factor.

Our convergence analysis is based on the eigenvalues of the iteration matrices corresponding to the two algorithms. We showed that the spectral radius plays a crucial role for the asymptotic convergence. The reason for slower asymptotic convergence of the RG algorithm lies in a larger spectral radius. This result shows the strength of such spectral methods and gives new insights in the behaviour of the TD(0) and the RG algorithm. We hope that the work presented in this paper will also be a starting point for more general theoretical results concerning the convergence speed of the TD(0) and the RG algorithm when linear function approximation is involved.

References

Baird, L. C. (1995). Residual algorithms: Reinforcement learning with function approximation. *Proceedings of the Twelfth International Conference on Machine Learning*. Morgan Kaufmann.

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro dynamic programming*. Athena Scientific.

Greenbaum, A. (1997). *Iterative methods for solving linear systems*, vol. 17 of *Frontiers in Applied Mathematics*. SIAM.

Horn, R. A., & Johnson, C. A. (1985). *Matrix analysis*. Cambridge University Press.

Schoknecht, R. (2003). Optimality of reinforcement learning algorithms with linear function approximation. *Advances in Neural Information Processing Systems 15*.

Schoknecht, R., & Merke, A. (2003). Convergent combinations of reinforcement learning with function approximation. *Advances in Neural Information Processing Systems 15*.

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

Sutton, R. S., Precup, D., & Singh, S. (1999). Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, *112*, 181–211.

Varga, R. S. (1962). *Matrix iterative analysis*. Prentice-Hall.

Appendix

Before we proof Theorem 1 we recall some common definitions and prove one auxiliary lemma.

For an arbitrary matrix $A \in \mathbb{C}^{n\times n}$ we denote by $\{\lambda_1(A),\ldots,\lambda_n(A)\} \subset \mathbb{C}$ the set of eigenvalues of A ordered such that $|\lambda_1(A)| \geq \ldots \geq |\lambda_n(A)|$. The Hermitian adjoint $A^\star$ of A is defined as the conjugate transpose of A, i.e. $A^\star = \bar{A}^\top$. In the case that A is a real matrix the Hermitian adjoint of A is just the transpose of A. The matrix $A^\star A$ is Hermitian (i.e. $(A^\star A) = (A^\star A)^\star$) and positive semidefinite and therefore all it's eigenvalues are nonnegative, $\{\lambda_1(A^\star A),\ldots,\lambda_n(A^\star A)\} \subset \mathbb{R}^{\geq 0}$. The square roots $\sigma_i(A) := \sqrt{\lambda_i(A^\star A)}$ of the eigenvalues of $A^\star A$ are also called singular values of A. A matrix norm $\|\cdot\|$ is a norm for which the usual norm axioms are satisfied, and additionally for arbitrary matrices $A, B \in \mathbb{C}^{n\times n}$ it holds that $\|AB\| \leq \|A\|\,\|B\|$. Now we are prepared for the following

Lemma 2 *Consider a nonsingular matrix $A \in \mathbb{C}^{n\times n}$. Let A have singular values $\sigma_1(A) \geq \ldots \geq \sigma_n(A) > 0$ and eigenvalues $\{\lambda_1(A),\ldots,\lambda_n(A)\} \subset \mathbb{C}$ ordered such that $|\lambda_1(A)| \geq \ldots \geq |\lambda_n(A)| > 0$. Then*

$$\frac{|\lambda_1(A)|}{|\lambda_n(A)|} \leq \frac{\sigma_1(A)}{\sigma_n(A)}.$$

Proof: The absolute value of the largest eigenvalue $\varrho(A) := |\lambda_1(A)|$ is also called the spectral radius of A. It is easily seen that $\varrho(A^{-1}) = \frac{1}{|\lambda_n(A)|}$. Furthermore $\varrho(A)$ has the advantageous property (cf. (Horn & Johnson, 1985)) that for every matrix norm $\|\cdot\|$ we have

$$\varrho(A) \leq \|A\|. \tag{16}$$

In the special case of the matrix norm $\|\cdot\|_2$ induced by the Euclidean vector norm, which according to (12) is defined as $\|A\|_2 = \max_{\|x\|_2=1}\|Ax\|_2$, the following holds (cf. (Horn & Johnson, 1985))

$$\|A\|_2 = \sqrt{\varrho(A^\star A)} = \sqrt{\varrho(AA^\star)}. \tag{17}$$

Putting it all together we obtain our assertion:

$$\begin{aligned}\frac{|\lambda_1(A)|}{|\lambda_n(A)|} &= \varrho(A)\varrho(A^{-1}) \overset{(16)}{\leq} \|A\|_2\|A^{-1}\|_2\\ &\overset{(17)}{=} \sqrt{\varrho(A^\star A)}\sqrt{\varrho((A^{-1})^\star A^{-1})}\\ &\overset{(17)}{=} \sqrt{\varrho(A^\star A)}\sqrt{\varrho(A^{-1}(A^{-1})^\star)}\\ &= \sqrt{\varrho(A^\star A)}\sqrt{\varrho((A^\star A)^{-1})} = \frac{\sigma_1(A)}{\sigma_n(A)}.\end{aligned}$$

□

Proof of Theorem 1

According to our assumptions $\Phi = I_N$ and $\widehat{D} = kI_N$. Without loss of generality we just consider the case $\widehat{D} = I_N$, the case $k > 1$ can be obtained by analogous calculations. Thus, from (7) and (9) we directly obtain

$$A_{TD} = \gamma\widehat{P} - I_N, \tag{18}$$

$$A_{RG} = -(\gamma\widehat{P} - I_N)^\top(\gamma\widehat{P} - I_N) = -A_{TD}^\top A_{TD}. \tag{19}$$

As usual, we denote the eigenvalues with the largest and smallest absolute values as $\lambda_1 := \lambda_1(A_{TD})$ and $\lambda_N := \lambda_N(A_{TD})$ respectively. It can be shown that the matrix A_{TD} defined according to (18) has only eigenvalues with negative real parts. A_{TD} and A_{RG} are therefore nonsingular. With the assumption that A_{TD} has only real eigenvalues we obtain $|\lambda_1| = -\lambda_1$, $|\lambda_N| = -\lambda_N$ and $\lambda_1 \leq \lambda_N < 0$. According to Corollary 1 the optimal learning rate α^*_{TD} is given by $\alpha^*_{TD} = \frac{2}{|\lambda_1|+|\lambda_N|}$. For the spectral radius of $(I_N + \alpha^*_{TD}A_{TD})$ we therefore obtain

$$\begin{aligned}\varrho(I_N + \alpha^*_{TD}A_{TD}) &= 1 + \alpha^*_{TD}\lambda_N = 1 - \frac{2|\lambda_N|}{|\lambda_1| + |\lambda_N|}\\ &= 1 - 2\left(1 + \frac{|\lambda_1|}{|\lambda_N|}\right)^{-1}.\end{aligned} \tag{20}$$

The matrix A_{RG} is symmetric and negative definite and therefore has only negative eigenvalues. Using the same arguments as for A_{TD} we conclude

$$\varrho(I_N + \alpha^*_{RG}A_{RG}) = 1 - 2\left(1 + \frac{\mu_1}{\mu_N}\right)^{-1}, \tag{21}$$

where μ_1 denotes the largest and μ_N is the smallest eigenvalue value of $A_{TD}^\top A_{TD}$. As all eigenvalues of $A_{TD}^\top A_{TD}$ are positive the square root can be taken. This yields the singular values of A_{TD}

$$\sigma_1 = \sqrt{\mu_1} \quad \text{and} \quad \sigma_N = \sqrt{\mu_N}. \tag{22}$$

As stated above, the matrices A_{TD} and A_{RG} are both nonsingular. Therefore, according to the arguments from section 3 the asymptotic convergence rate is determined by the corresponding spectral radius, where smaller spectral radius implies faster convergence and vice versa. Thus, looking at (20) and (21) it remains to show that

$$1 - 2\left(1 + \frac{|\lambda_1|}{|\lambda_N|}\right)^{-1} \leq 1 - 2\left(1 + \frac{\mu_1}{\mu_N}\right)^{-1}$$

or equivalently that $|\lambda_1|/|\lambda_N| \leq \mu_1/\mu_N$. Using (22) this is equivalent to showing that $|\lambda_1|/|\lambda_N| \leq (\sigma_1/\sigma_N)^2$. From Lemma 2 we already know that $|\lambda_1|/|\lambda_N| \leq \sigma_1/\sigma_N$. Because $\sigma_1/\sigma_N \geq 1$ we can extend this to $|\lambda_1|/|\lambda_N| \leq \sigma_1/\sigma_N \leq (\sigma_1/\sigma_N)^2$. Equality is only possibly if $\sigma_1/\sigma_N = 1$, i.e. if all eigenvalues of A_{RG} are identical. In this case both algorithms converge equally fast asymptotically. In all other cases we obtain $|\lambda_1|/|\lambda_N| \leq \sigma_1/\sigma_N < (\sigma_1/\sigma_N)^2$ due to $\sigma_1/\sigma_N > 1$. Thus, the TD(0) algorithm converges asymptotically faster than the RG algorithm. □