Approximate Dynamic Programming for Two-Player Zero-Sum Markov Games

Julien Perolat[(1)] JULIEN.PEROLAT@ED.UNIV-LILLE1.FR
Bruno Scherrer[(2)] BRUNO.SCHERRER@INRIA.FR
Bilal Piot[(1)] BILAL.PIOT@UNIV-LILLE3.FR
Olivier Pietquin[(1,3)] OLIVIER.PIETQUIN@UNIV-LILLE1.FR
[(1)]Univ. Lille, CRIStAL, SequeL team, France
[(2)]Inria, Villers-lès-Nancy, F-54600, France
[(3)]Institut Universitaire de France (IUF), France

Abstract

This paper provides an analysis of error propagation in Approximate Dynamic Programming applied to zero-sum two-player Stochastic Games. We provide a novel and unified error propagation analysis in L_p-norm of three well-known algorithms adapted to Stochastic Games (namely Approximate Value Iteration, Approximate Policy Iteration and Approximate Generalized Policy Iteratio,n). We show that we can achieve a stationary policy which is $\frac{2\gamma\epsilon+\epsilon'}{(1-\gamma)^2}$-optimal, where ϵ is the value function approximation error and ϵ' is the approximate greedy operator error. In addition, we provide a practical algorithm (AGPI-Q) to solve infinite horizon γ-discounted two-player zero-sum Stochastic Games in a batch setting. It is an extension of the Fitted-Q algorithm (which solves Markov Decisions Processes from data) and can be non-parametric. Finally, we demonstrate experimentally the performance of AGPI-Q on a simultaneous two-player game, namely Alesia.

1. Introduction

A wide range of complex problems (*e.g.* computer networks, human-computer interfaces, games as chess or checkers) can be addressed as multi-agent systems. This is why Multi-Agent Reinforcement Learning (MARL) (Busoniu et al., 2008) has received a growing interest in the last few years. While, in decision theory, Markov Decision Processes (MDPs) (Puterman, 1994) are widely used to control a single agent in a complex environment, Markov Games (MGs) (also named Stochastic Games (SGs)) (Shapley, 1953) are an extension of this formal framework to describe multi-agent systems. As for MDPs, SGs constitute a model for MARL (Littman, 1994) and have been largely studied in past years (Hu & Wellman, 2003; Greenwald et al., 2003; Bowling & Veloso, 2001). In some sense, SGs are a generalization of both the game-theory and the MDP frameworks where the agents may have different pay-offs (rewards in the MDP vocabulary) they aim at maximizing.

This paper contributes to solving large scale games with unknown dynamics. Especially, it addresses the problem of finding the Nash equilibrium for infinite horizon γ-discounted two-player zero-sum SGs in an approximate fashion, possibly from batch data. SGs are modeled as extensions of MDPs where Dynamic Programming (DP) commonly serves as a basis to a wide range of practical solutions (Puterman, 1994). However, when the size of the game is too large or when its dynamics is not perfectly known, exact solutions cannot be computed. In that case, Approximate Dynamic Programming (ADP) approaches are preferred as for MDPs (Bertsekas, 1995). Because DP relies on an iterative schema, interleaving Value Function Approximation and Greedy Operator Approximations, two types of errors (corresponding to each source of approximation) are introduced and can accumula,te over iterationsprint("—

More specifically, this paper provides a theoretical error propagation analysis in L_p-norm of well-known ADP algorithms applied to games such as Approximate Value Iteration (AVI), Approximate Policy Iteration (API) and Approximate Generalized Policy Iteration (AGPI) (described later). It also proposes a novel algorithm, named Approximate Generalized Policy Iteration-Q (AGPI-Q) extending Fitted-Q iteration (Ernst et al., 2005) which is further analyzed in terms of error and complexity. AGPI-Q is evaluated on a simultaneous two-player game, namely Alesia.

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

1.1. Dynamic Programming techniques for MDP

ADP for MDPs has been the topic of many studies these last two decades. Bounds in L_∞ can be found in (Bertsekas, 1995) while L_p-norm ones were published in (Munos & Szepesvári, 2008) and (Farahmand et al., 2010). Because approximations often come from the use of supervised-learning algorithms, L_p-norm bounds are a significant improvement over L_∞ ones. Indeed, they can rely on upper bounds provided in the supervised learning literature to estimate the overall error on the learnt policy. For a unified analysis of PI, VI and MPI in L_p-norm see Scherrer et al. (2012). It allows evaluating several practical algorithms like Fitted-Q iteration (Ernst et al., 2005; Antos et al., 2008), Classification-based MPI (Lagoudakis & Parr, 2003b; Lazaric et al., 2010; Gabillon et al., 2011) and LSPI (Lagoudakis & Parr, 2003a).

1.2. Dynamic Programming techniques for Stochastic Games

Zero-sum two-player γ-discounted SGs are quite close to γ-discounted MDPs. In fact, the development of DP techniques for SGs has followed closely the development of DP for MDPs (Shapley, 1953). Most of these algorithms are detailed in Patek (1997) in the framework of Stochastic Shortest Path Games (SSPG). Zero-sum two-player discounted SG is a subclass of SSPG. Similar algorithms exist to solve those games including PI (Patek, 1997), VI (Shapley, 1953) and MPI (called also Generalized Policy Iteration (GPI) (Patek, 1997) in the context of SSPG). Yet, there has been very little attention paid to ADP for SGs after Patek's work except from Lagoudakis & Parr (2002). To our knowledge, no analyses of the approximate version of those algorithms exist in L_p-norm.

2. Framework

A SG is a generalization of an MDP to a n-player setting. Each player has a control on the SG through a set of actions. At each step of the game, all players simultaneously choose an action. The reward each player gets after one step depends on the state and on the joint action of all players. Furthermore, the dynamics of the SG may depend on the state and on the actions of all players.

Each player is interested in maximizing some criterion on this sequence of rewards. Here, we will consider the problem of computing the minimax equilibrium in a two-player zero-sum SG where each player tries to maximize his own value, defined as the expected γ-discounted cumulative reward. The value attained at the minimax equilibrium is the optimal value. In the case of simultaneous games, the major difference between those games and MDPs is that each player may have to randomize his strategies to reach this equilibrium (Von Neumann, 1947).

2.1. Two-Player Zero-Sum Discounted Stochastic Games

A two-player γ-discounted SG can be considered as a tuple $(S, (A^1(s))_{s\in S}, (A^2(s))_{s\in S}, p, r, \gamma)$ where S is the state space[1], $A^1(s)$ is the finite set of actions player 1 can play in state s, $A^2(s)$ is the finite set of actions player 2 can play in state s, $p(s'|s, a^1, a^2)$ with $a^1 \in A^1(s)$ and $a^2 \in A^2(s)$ is the probability transition from state s to state s', $r(s, a^1, a^2)$ with $a^1 \in A^1(s)$ and $a^2 \in A^2(s)$ is the reward of both players bounded by $R_{\max}$ and γ is the discount factor. A strategy π^i, where $i \in \{1, 2\}$, maps a state $s \in S$ to a probability distribution $\pi^i(.|s)$ over $A^i(s)$. Those strategies are named policies in the MDP literature. We will adopt both vocabulary indifferently.

The stochastic kernel characterising the transition of the game is $\mathcal{P}_{\pi^1,\pi^2}(s'|s) = E_{a^1\sim\pi^1(.|s),a^2\sim\pi^2(.|s)}[p(s'|s, a^1, a^2)]$. The stochastic kernel is the probability to go from s to s' when player 1 is playing strategy π^1 and player 2 is playing strategy π^2. The reward function of strategies (π^1, π^2) will be noted $r_{\pi^1,\pi^2} = E_{a^1\sim\pi^1(.|s),a^2\sim\pi^2(.|s)}[r(s, a^1, a^2)]$. This quantity represents the reward each player can expect while player 1 plays π^1 and player 2 plays π^2.

One can see zero-sum two-player games as two players maximizing opposite rewards. Another way to see it is that the goal of player 1 is to maximize his cumulated γ-discounted reward while the goal of player 2 is to minimize it. From now on, we will note μ the policy of the maximizer and ν the policy of the minimizer.

Let $v_{\mu,\nu}(s) = E[\sum_{t=0}^{+\infty} \gamma^t r_{\mu,\nu}(s_t)|s_0 = s, s_{t+1} \sim \mathcal{P}_{\mu,\nu}(.|s_t)]$ be the cumulative reward if the players 1 and 2 respectively use the stationary strategies μ and ν. The value function $v_{\mu,\nu}$ maps the state s to its value $v_{\mu,\nu}(s)$.

2.2. Bellman Operators

Let us define the following five operators on value v:

$$\mathcal{T}_{\nu,\mu}\, v = r_{\mu,\nu} + \gamma \mathcal{P}_{\mu,\nu}\, v$$

$$\mathcal{T}_\mu\, v = \min_\nu \mathcal{T}_{\mu,\nu}\, v \qquad \hat{\mathcal{T}}_\nu\, v = \max_\mu \mathcal{T}_{\mu,\nu}\, v$$

$$\mathcal{T}\, v = \max_\mu \mathcal{T}_\mu\, v \qquad \hat{\mathcal{T}}\, v = \min_\nu \hat{\mathcal{T}}_\nu\, v$$

Here μ and ν are random policies. The $\min$ and $\max$ are well defined because μ and ν lie in compact sets since we are considering finite sets of actions. All those operators are contractions in L_∞-norm with constant γ. We have the

[1] We will only consider a finite state space. The case where the state space is continuous is beyond the scope of this paper.

following remarkable property (Patek, 1997):

$$\forall v,\ \hat{\mathcal{T}}\ v = \mathcal{T}\ v. \tag{1}$$

Equation (1) is a consequence of von Neumann's Minimax theorem (Von Neumann, 1947; Patek, 1997).

2.3. Minimax Equilibrium

In the setting of zero-sum two-player SGs, notions of minimax equilibrium and of Nash equilibrium are equivalent. In this case, the optimal value of the game is:

$$v^* = \min_{\nu} \max_{\mu} v_{\mu,\nu} = \max_{\mu} \min_{\nu} v_{\mu,\nu}. \tag{2}$$

The existence of v^* is proved in Patek (1997) for a more general class of games. This value can be achieved using mixed stationary strategies (Patek, 1997). Equation (2) is a consequence of Equation (1) and of the contraction property. The value v^* is the unique fixed point of the operator $\mathcal{T}$. We will note $v_\mu = \inf_\nu v_{\mu,\nu}$ the fixed point of operator $\mathcal{T}_\mu$. An optimal counter strategy against μ is any strategy ν satisfying $v_{\mu,\nu} = v_\mu$.

2.4. Policy Iteration, Value Iteration and Generalized Policy Iteration

The three algorithms we intend to analyse are VI, PI and GPI. They all share the same greedy step. A strategy μ is greedy with respect to some value v (noted $\mu \in \mathcal{G}(v)$) when $\mathcal{T}\ v = \mathcal{T}_\mu\ v = \min_\nu \mathcal{T}_{\mu,\nu}\ v$.

Remark 1. *In practice, trying to find μ greedy with respect to a value v means trying to maximize $(\mathcal{T}_\mu\ v)(s)$ for each state s. Let us fix some state s; we try to find*

$$(\max_{\mu} \min_{\nu} \mathcal{T}_{\mu,\nu}\ v)(s) =$$
$$\max_{\mu(.|s)} \min_{\nu(.|s)} E_{\substack{a\sim\mu(.|s),a'\sim\nu(.|s),\\ \Sigma\sim\mathcal{P}_{\mu,\nu}\ (.|s)}} [r(s,a,a') + \gamma v(\Sigma)].$$

For a constant state s, this is equivalent to trying to find the minimax equilibrium of a matrix game. The matrix of the game with stochastic reward is defined by $(r(s,a,a') + \gamma v(\Sigma_{a,a'}))_{a\in A^1(s),a'\in A^2(s)}$ where $\Sigma_{a,a'} \sim p(.|s,a,a')$. From the expectation of this matrix, the minimax equilibrium can be computed with linear programming.

In the case of turn-based games, finding a greedy policy is much simpler. Indeed, since at each state only one player controls the SG, finding a greedy strategy is reduced to finding a maximum over the actions of player 1.

The algorithms differ by the way they do the evaluation step. We describe them by showing how they work from iteration k to iteration $k+1$. **Value Iteration** (VI) iterates as follows:

$$v_{k+1} = \mathcal{T}\ v_k.$$

This can equivalently be written as follows:

$$\mu_{k+1} = \mathcal{G}(v_k),$$
$$v_{k+1} = \mathcal{T}_{\mu_{k+1}}\ v_k.$$

Policy Iteration (PI) iterates as follows

$$\mu_{k+1} = \mathcal{G}(v_k),$$
$$v_{k+1} = v_{\mu_{k+1}} = (\mathcal{T}_{\mu_{k+1}}\ )^{+\infty} v_k.$$

Finally, **Generalized Policy Iteration** (GPI) iterates in a way that interpolates between VI and PI:

$$\mu_{k+1} = \mathcal{G}(v_k),$$
$$v_{k+1} = (\mathcal{T}_{\mu_{k+1}}\ )^{m} v_k.$$

It is clear that GPI generalizes VI and PI. In particular, the error propagation bounds for VI (when $m = 1$) and PI (when $m = \infty$) will follow from the analysis we shall provide for GPI.

2.5. Example and Special Cases

In this paper, we consider the game Alesia (Meyer et al., 1997), that is a two-player zero-sum simultaneous game. Both players control a token in the center of a five box board. At the beginning each player has a finite budget n. At each turn, every one has to bet at least 1 if his own remaining budget is not 0. The goal of each player is to put the token on his side of the board. At each turn the token moves toward the objective of the player who bets the most (in case of equality the token doesn't move). The one wining is the one who can push his token outside of the board.

This game can be modelled as follows. The state space S is a triplet (token position $\in \{1,...,5\}$, budget of player 1 $\in \{0,...,n\}$, budget of player 2 $\in \{0,...,n\}$) and an absorbing state Ω where the players fall at the end of the game. The action space in state (s,u,v) is $A^1((s,u,v))$ for player 1. If $u \neq 0$, then $A^1((s,u,v)) = \{1,...,u\}$ else $A^1((s,u,v)) = \{0\}$. For player 2, if $v \neq 0$, then $A^2((s,u,v)) = \{1,...,v\}$ else $A^1((s,u,v)) = \{0\}$. If player 1 bets a_1 and player 2 bets a_2 in state (s,u,v), then the next state is $(s', u - a_1, v - a_2)$ where $s' = s - \mathbb{1}_{a_1 \leq a_2} + \mathbb{1}_{a_2 \leq a_1}$ or Ω if $s' \notin \{1,...,5\}$. The reward is 1 if the token leaves the board from position 5 and -1 if the token leaves the board from position 1.

Remark 2. *In turn-based games, each state is controlled by a single player. In the zero-sum two-player SGs framework, they can be seen as a game where $\forall s \in S,\ card(A^1(s)) = 1$ or $card(A^2(s)) = 1$. In this special case, optimal strategies are deterministic (Hansen et al., 2013). Furthermore, and as we already mentionned, the greedy step (see definition in Sec. 2.4) reduces to finding*

a maximum rather than a minimax equilibrium and is thus significantly simpler. Furthermore, one can see an MDP as a zero-sum two-player SG where one player has no influence on both the reward and the dynamics. Therefore, our analysis should be consistent with previous MDP analyses.

3. Stochastic Games and Approximate Generalized Policy Iteration (AGPI)

In this section, we analyse the algorithm in the case of approximations in the greedy and evaluation steps. This analysis was done in L_∞-norm in Patek (1997) for PI. We generalize it for AGPI in the case of σ-weighted L_p-norm, that is defined for a function h and a distribution σ on the state space as $\|h\|_{p,\sigma} = \left(\sum_{s\in S} |h(s)|^p \sigma(s)\right)^{\frac{1}{p}}$.

3.1. Approximate Generalized Policy Iteration

In Patek (1997), this algorithm is presented as GPI. It has been first presented in (Van Der Wal, 1978). Similarly to its exact counterpart, an iteration of this algorithm can be divided in two steps: a greedy step and an evaluation step. The main difference is that we account for possible errors in both steps (respectively ϵ'_k and ϵ_k).

For the **greedy step**, we shall write $\mu_k \leftarrow \hat{\mathcal{G}}_{\epsilon'_k}(v_{k-1})$ for:

$$\begin{aligned} \mathcal{T}\ v_{k-1} &\leq \mathcal{T}_{\mu_k}\ v_{k-1} + \epsilon'_k \\ \text{or, } \forall \mu\ \mathcal{T}_\mu\ v_{k-1} &\leq \mathcal{T}_{\mu_k}\ v_{k-1} + \epsilon'_k. \end{aligned} \tag{3}$$

In other words, the strategy μ_k is not necessarily the best strategy, but it has to be at most ϵ'_k away from the best strategy.

For the **evaluation step**, we consider that we may have an additive error ϵ_k:

$$v_k = (\mathcal{T}_{\mu_k})^m v_{k-1} + \epsilon_k. \tag{4}$$

Remark 3. *Evaluation step: In the evaluation step the policy μ_k is fixed and we apply m times the operator $\mathcal{T}_{\mu_k} \cdot = \min_\nu \mathcal{T}_{\mu_k,\nu} \cdot$. Since the policy of the maximizer is fixed, the problem solved in the evaluation step consists in finding an optimal m-horizon counter-policy for the minimizer.*

3.2. Error Propagation

Since errors may accumulate from iterations to iterations, we are interested in bounding the difference

$$l_k = v_* - v_{\mu_k} \geq 0,$$

where v_* is the minimax value of the game (obtained when both players play the Nash equilibrium μ_* and ν_*) and where v_{μ_k} is the value when the maximizer plays μ_k and the minimizer plays the optimal counter-strategy against μ_k. This is a natural measure of quality for the strategy μ_k that would be output by the approximate algorithm.

By definition, we have:

$$\forall \nu, \forall v,\ \mathcal{T}_{\mu_k}\ v \leq \mathcal{T}_{\mu_k,\nu}\ v. \tag{5}$$

In addition, we shall consider a few notations. The minimizer policies ν_k^i, $\tilde{\nu}_k$, $\hat{\nu}_k$ and $\overline{\nu}_k$ are policies that respectively satisfy:

$$(\mathcal{T}_{\mu_k})^{i+1} v_{k-1} = \mathcal{T}_{\mu_k,\nu_k^i} \ldots \mathcal{T}_{\mu_k,\nu_k^1}\ \mathcal{T}_{\mu_k}\ v_{k-1}, \tag{6}$$

$$\mathcal{T}_{\mu_*}\ v_k = \mathcal{T}_{\mu_*,\tilde{\nu}_k}\ v_k, \tag{7}$$

$$\mathcal{T}_{\mu_k}\ v_k = \mathcal{T}_{\mu_k,\hat{\nu}_k}\ v_k,$$

$$\mathcal{T}_{\mu_k}\ v_{\mu_k} = \mathcal{T}_{\mu_k,\overline{\nu}_k}\ v_{\mu_k}. \tag{8}$$

In order to bound l_k, we will study the following quantities similar to those introduced by Scherrer et al. (2012) (recall that $\mathcal{T}_\mu$ and $\mathcal{T}_{\mu,\nu}$ are defined in Sec. 2.2 and the stochastic kernel is defined in Sec. 2.1):

$$\begin{aligned} d_k &= v_* - (\mathcal{T}_{\mu_k})^m v_{k-1} = v_* - (v_k - \epsilon_k), \\ s_k &= (\mathcal{T}_{\mu_k})^m v_{k-1} - v_{\mu_k} = (v_k - \epsilon_k) - v_{\mu_k}, \\ b_k &= v_k - \mathcal{T}_{\mu_{k+1}}\ v_k, \\ x_k &= (\mathcal{I} - \gamma \mathcal{P}_{\mu_k,\hat{\nu}_k})\epsilon_k + \epsilon'_{k+1}, \\ y_k &= -\gamma \mathcal{P}_{\mu_*,\tilde{\nu}_k}\ \epsilon_k + \epsilon'_{k+1}. \end{aligned}$$

Notice that $l_k = d_k + s_k$. We shall prove the following relations, similar to the one proved in (Scherrer et al., 2012).

Lemma 1. *The following linear relations hold:*

$$b_k \leq \gamma \mathcal{P}_{\mu_k,\hat{\nu}_k}\ \gamma \mathcal{P}_{\mu_k,\nu_k^{m-1}} \ldots \gamma \mathcal{P}_{\mu_k,\nu_k^1}\ b_{k-1} + x_k,$$

$$d_{k+1} \leq \gamma \mathcal{P}_{\mu_*,\tilde{\nu}_k}\ d_k + y_k + \sum_{j=1}^{m-1} \gamma \mathcal{P}_{\mu_k,\nu_k^j} \ldots \gamma \mathcal{P}_{\mu_k,\nu_k^1}\ b_k,$$

$$s_k \leq (\gamma \mathcal{P}_{\mu_k,\overline{\nu}_k})^m \Big(\sum_{i=1}^{\infty} \gamma \mathcal{P}_{\mu_k,\nu_k^i} \ldots \gamma \mathcal{P}_{\mu_k,\nu_k^1}\ b_{k-1}\Big).$$

Contrary to the analysis of Scherrer et al. (2012) for MDPs in which the operator $\mathcal{T}_\mu$ is affine, it is in our case nonlinear. The proof that we now develop is thus slightly trickier.

Proof. **Let us start with b_k:**

$$\begin{aligned} b_k &= v_k - \mathcal{T}_{\mu_{k+1}}\ v_k, \\ &= v_k - \mathcal{T}_{\mu_k}\ v_k + \mathcal{T}_{\mu_k}\ v_k - \mathcal{T}_{\mu_{k+1}}\ v_k. \end{aligned}$$

In Equation (3) with $\mu = \mu_{k-1}$ and $k \leftarrow k+1$, we have $\mathcal{T}_{\mu_k}\ v_k \leq \mathcal{T}_{\mu_{k+1}}\ v_k + \epsilon'_{k+1}$ then:

$$\begin{aligned}
b_k &\leq v_k - \mathcal{T}_{\mu_k}\ v_k + \epsilon'_{k+1},\\
&= v_k - \epsilon_k - \underbrace{\mathcal{T}_{\mu_k}\ v_k}_{=\mathcal{T}_{\mu_k,\hat{\nu}_k}\ v_k} + \gamma \mathcal{P}_{\mu_k,\hat{\nu}_k}\ \epsilon_k\\
&\quad + \epsilon_k - \gamma \mathcal{P}_{\mu_k,\hat{\nu}_k}\ \epsilon_k + \epsilon'_{k+1},\\
&= v_k - \epsilon_k - \underbrace{\mathcal{T}_{\mu_k,\hat{\nu}_k}\ (v_k - \epsilon_k)}_{\mathcal{T}_{\mu_k,\hat{\nu}_k} \text{ is affine}}\\
&\quad + (\mathcal{I} - \gamma \mathcal{P}_{\mu_k,\hat{\nu}_k}\ )\epsilon_k + \epsilon'_{k+1}.
\end{aligned}$$

From Equation (4), we have $v_k - \epsilon_k = (\mathcal{T}_{\mu_k}\ )^m v_{k-1}$. Thus,

$$\begin{aligned}
b_k &\leq (\mathcal{T}_{\mu_k}\ )^m v_{k-1} - \mathcal{T}_{\mu_k,\hat{\nu}_k}\ (\mathcal{T}_{\mu_k}\ )^m v_{k-1} + x_k \quad \text{(Eq. (6))},\\
&= (\mathcal{T}_{\mu_k}\ )^m v_{k-1}\\
&- \mathcal{T}_{\mu_k,\hat{\nu}_k}\ \mathcal{T}_{\mu_k,\nu_k^{m-1}}\ ...\mathcal{T}_{\mu_k,\nu_k^1}\ (\mathcal{T}_{\mu_k}\ v_{k-1}) + x_k \quad \text{(Eq. (6))},\\
&\leq \mathcal{T}_{\mu_k,\hat{\nu}_k}\ \mathcal{T}_{\mu_k,\nu_k^{m-1}}\ ...\mathcal{T}_{\mu_k,\nu_k^1}\ v_{k-1}\\
&- \mathcal{T}_{\mu_k,\hat{\nu}_k}\ \mathcal{T}_{\mu_k,\nu_k^{m-1}}\ ...\mathcal{T}_{\mu_k,\nu_k^1}\ (\mathcal{T}_{\mu_k}\ v_{k-1}) + x_k \quad \text{(Eq. (5))},\\
&= \gamma \mathcal{P}_{\mu_k,\hat{\nu}_k}\ \gamma \mathcal{P}_{\mu_k,\nu_k^{m-1}}\ ...\gamma \mathcal{P}_{\mu_k,\nu_k^1}\ (v_{k-1} - \mathcal{T}_{\mu_k}\ v_{k-1})\\
&+ x_k,\\
&\leq \gamma \mathcal{P}_{\mu_k,\hat{\nu}_k}\ \gamma \mathcal{P}_{\mu_k,\nu_k^{m-1}}\ ...\gamma \mathcal{P}_{\mu_k,\nu_k^1}\ b_{k-1} + x_k.
\end{aligned}$$

To bound d_{k+1}, we decompose it in the three following terms:

$$\begin{aligned}
d_{k+1} &= v_* - (\mathcal{T}_{\mu_{k+1}}\ )^m v_k,\\
&= \underbrace{\mathcal{T}_{\mu_*}\ v_* - \mathcal{T}_{\mu_*}\ v_k}_{①} + \underbrace{\mathcal{T}_{\mu_*}\ v_k - \mathcal{T}_{\mu_{k+1}}\ v_k}_{②}\\
&\quad + \underbrace{\mathcal{T}_{\mu_{k+1}}\ v_k - (\mathcal{T}_{\mu_{k+1}}\ )^m v_k}_{③}.
\end{aligned}$$

In this equation, term ① can be upper-bounded as follows:

$$\begin{aligned}
&\mathcal{T}_{\mu_*}\ v_* - \mathcal{T}_{\mu_*}\ v_k\\
&= \mathcal{T}_{\mu_*}\ v_* - \mathcal{T}_{\mu_*,\tilde{\nu}_k}\ v_k \text{ with } \tilde{\nu}_k \text{ defined in Eq. (7)},\\
&\leq \mathcal{T}_{\mu_*,\tilde{\nu}_k}\ v_* - \mathcal{T}_{\mu_*,\tilde{\nu}_k}\ v_k \text{ since } \forall \nu,\ \mathcal{T}_{\mu_*}\ . \leq \mathcal{T}_{\mu_*,\nu}\ .\\
&= \gamma \mathcal{P}_{\mu_*,\tilde{\nu}_k}\ (v_* - v_k).
\end{aligned}$$

By definition ② is bounded by the greedy error:

$$\mathcal{T}_{\mu_*}\ v_k - \mathcal{T}_{\mu_{k+1}}\ v_k \leq \epsilon'_{k+1} \text{(3) with } \mu \leftarrow \mu_*, k \leftarrow k+1$$

Finally, bounding term ③ involves the b_k quantity:

$$\begin{aligned}
&\mathcal{T}_{\mu_{k+1}}\ v_k - (\mathcal{T}_{\mu_{k+1}}\ )^m v_k,\\
&= \sum_{j=1}^{m-1} (\mathcal{T}_{\mu_{k+1}}\ )^j v_k - (\mathcal{T}_{\mu_{k+1}}\ )^{j+1} v_k,\\
&= \sum_{j=1}^{m-1} (\mathcal{T}_{\mu_{k+1}}\ )^j v_k - \mathcal{T}_{\mu_{k+1},\nu_{k+1}^j}\ ...\mathcal{T}_{\mu_{k+1},\nu_{k+1}^1}\ \mathcal{T}_{\mu_k+1}\ v_k,\\
&\leq \sum_{j=1}^{m-1} [\mathcal{T}_{\mu_{k+1},\nu_{k+1}^j}\ ...\mathcal{T}_{\mu_{k+1},\nu_{k+1}^1}\ v_k\\
&\qquad - \mathcal{T}_{\mu_{k+1},\nu_{k+1}^j}\ ...\mathcal{T}_{\mu_{k+1},\nu_{k+1}^1}\ \mathcal{T}_{\mu_k+1}\ v_k] \text{ see (5)},\\
&= \sum_{j=1}^{m-1} \gamma \mathcal{P}_{\mu_{k+1},\nu_{k+1}^j}\ ...\gamma \mathcal{P}_{\mu_{k+1},\nu_{k+1}^1}\ (v_k - \mathcal{T}_{\mu_k+1}\ v_k),\\
&= \sum_{j=1}^{m-1} \gamma \mathcal{P}_{\mu_{k+1},\nu_{k+1}^j}\ ...\gamma \mathcal{P}_{\mu_{k+1},\nu_{k+1}^1}\ b_k.
\end{aligned}$$

Then d_{k+1} becomes:

$$\begin{aligned}
d_{k+1} &\leq \gamma \mathcal{P}_{\mu_*,\tilde{\nu}_k}\ (v_* - v_k) + \epsilon'_{k+1}\\
&\quad + \sum_{j=1}^{m-1} \gamma \mathcal{P}_{\mu_{k+1},\nu_{k+1}^j}\ ...\gamma \mathcal{P}_{\mu_{k+1},\nu_{k+1}^1}\ b_k,\\
&\leq \gamma \mathcal{P}_{\mu_*,\tilde{\nu}_k}\ (v_* - v_k) + \gamma \mathcal{P}_{\mu_*,\tilde{\nu}_k}\ \epsilon_k - \mathcal{P}_{\mu_*,\tilde{\nu}_k}\ \epsilon_k\\
&\quad + \epsilon'_{k+1} + \sum_{j=1}^{m-1} \gamma \mathcal{P}_{\mu_{k+1},\nu_{k+1}^j}\ ...\gamma \mathcal{P}_{\mu_{k+1},\nu_{k+1}^1}\ b_k,\\
&\leq \gamma \mathcal{P}_{\mu_*,\tilde{\nu}_k}\ (v_* - (\underbrace{v_k - \epsilon_k}_{(\mathcal{T}_{\mu_k}\ )^m v_{k-1}}))\underbrace{-\mathcal{P}_{\mu_*,\tilde{\nu}_k}\ \epsilon_k + \epsilon'_{k+1}}_{y_k}\\
&\quad + \sum_{j=1}^{m-1} \gamma \mathcal{P}_{\mu_{k+1},\nu_{k+1}^j}\ ...\gamma \mathcal{P}_{\mu_{k+1},\nu_{k+1}^1}\ b_k,\\
&\leq \gamma \mathcal{P}_{\mu_*,\tilde{\nu}_k}\ d_k + y_k\\
&\quad + \sum_{j=1}^{m-1} \gamma \mathcal{P}_{\mu_{k+1},\nu_{k+1}^j}\ ...\gamma \mathcal{P}_{\mu_{k+1},\nu_{k+1}^1}\ b_k.
\end{aligned}$$

Let us finally bound s_k:

$$\begin{aligned}
s_k &= (\mathcal{T}_{\mu_k}\ )^m v_{k-1} - v_{\mu_k},\\
&= (\mathcal{T}_{\mu_k}\ )^m v_{k-1} - (\mathcal{T}_{\mu_k}\ )^\infty v_{k-1},\\
&= (\mathcal{T}_{\mu_k}\ )^m v_{k-1} - (\mathcal{T}_{\mu_k}\ )^m (\mathcal{T}_{\mu_k}\ )^\infty v_{k-1},\\
&= (\mathcal{T}_{\mu_k}\ )^m v_{k-1} - (\mathcal{T}_{\mu_k,\overline{\nu}_k}\ )^m (\mathcal{T}_{\mu_k}\ )^\infty v_{k-1},\\
&\quad \text{with } \overline{\nu}_k \text{ defined in (8)}\\
&\leq (\mathcal{T}_{\mu_k,\overline{\nu}_k}\ )^m v_{k-1} - (\mathcal{T}_{\mu_k,\overline{\nu}_k}\ )^m (\mathcal{T}_{\mu_k}\ )^\infty v_{k-1} \text{ since (5)},\\
&= (\gamma \mathcal{P}_{\mu_k,\overline{\nu}_k}\ )^m (v_{k-1} - (\mathcal{T}_{\mu_k}\ )^\infty v_{k-1}),\\
&= (\gamma \mathcal{P}_{\mu_k,\overline{\nu}_k}\ )^m (\sum_{i=0}^{\infty} (\mathcal{T}_{\mu_k}\ )^i v_{k-1} - (\mathcal{T}_{\mu_k}\ )^i (\mathcal{T}_{\mu_k}\ v_{k-1})),\\
&\leq (\gamma \mathcal{P}_{\mu_k,\overline{\nu}_k}\ )^m \sum_{i=0}^{\infty} \gamma \mathcal{P}_{\mu_k,\nu_k^i}\ ...\gamma \mathcal{P}_{\mu_k,\nu_k^1}\ (v_{k-1} - \mathcal{T}_{\mu_k}\ v_{k-1}),\\
&\leq (\gamma \mathcal{P}_{\mu_k,\overline{\nu}_k}\ )^m \sum_{i=0}^{\infty} \gamma \mathcal{P}_{\mu_k,\nu_k^i}\ ...\gamma \mathcal{P}_{\mu_k,\nu_k^1}\ b_{k-1}. \qquad \square
\end{aligned}$$

From linear recursive relations of the kind of Lemma 1, Scherrer et al. (2012) show how to deduce a bound on the

L_p-norm of l_k. This part of the proof being identical to that of Scherrer et al. (2012), we do not develop it here. For completeness however, we include it in Appendix A.1 of the Supplementary Material.

Theorem 1. *Let ρ and σ be distributions over states. Let p, q and q' be such that $\frac{1}{q}+\frac{1}{q'}=1$. Then, after k iterations, we have:*

$$\begin{aligned}\|l_k\|_{p,\rho} \leq & \frac{2(\gamma-\gamma^k)(\mathcal{C}_q^{1,k,0})^{\frac{1}{p}}}{(1-\gamma)^2} \sup_{1\leq j\leq k-1} \|\epsilon_j\|_{pq',\sigma}, \\ & + \frac{(1-\gamma^k)(\mathcal{C}_q^{0,k,0})^{\frac{1}{p}}}{(1-\gamma)^2} \sup_{1\leq j\leq k} \|\epsilon'_j\|_{pq',\sigma}, \\ & + \frac{2\gamma^k}{1-\gamma}(\mathcal{C}_q^{k,k+1,0})^{\frac{1}{p}} \min(\|d_0\|_{pq',\sigma}, \|b_0\|_{pq',\sigma}).\end{aligned}$$

where

$$\mathcal{C}_q^{l,k,d} = \frac{(1-\gamma)^2}{\gamma^l-\gamma^k}\sum_{i=l}^{k-1}\sum_{j=i}^{\infty}\gamma^j c_q(j+d),$$

with the following norm of a Radon-Nikodym derivative:

$$c_q(j) = \sup_{\mu_1,\nu_1,\ldots,\mu_j,\nu_j} \left\| \frac{d(\rho\mathcal{P}_{\mu_1,\nu_1}\ldots\mathcal{P}_{\mu_j,\nu_j})}{d\sigma} \right\|_{q,\sigma}.$$

Remark 4. *The Radon-Nikodym derivative of measure ρ on S with respect to measure σ on S is in the discrete case the function h defined by: $h(s)=\frac{\rho(s)}{\sigma(s)}$ when $\sigma(s)\neq 0$, ∞ otherwise.*

Remark 5. *If player 2 has no influence on the game, then the γ-discounted two-player zero-sum SG is simply a γ-discounted MDP. In that case the bound is the same as in Scherrer et al. (2012).*

Remark 6. *In the case of a SG with no discount factor but with an absorbing state the expression given in lemma 1 is still valid. Instead of having a γ-discounted transition kernel we would have a simple transition kernel. And if this transition kernel has the following property*

$$\exists l, \quad \sup_{\mu_0,\nu_0,\ldots,\mu_l,\nu_l} \left\| \prod_{i=0}^{l}\mathcal{P}_{\mu_i,\nu_i} \right\|_{\infty} \leq \gamma < 1,$$

we could still have upper bounds on the propagation of errors.

Remark 7. *One should notice that when p tends to infinity, the bound becomes:*

$$\liminf_{k\to+\infty}\|l_k\|_{\infty} \leq \frac{2\gamma}{(1-\gamma)^2}\epsilon + \frac{1}{(1-\gamma)^2}\epsilon',$$

where ϵ and ϵ' are respectively the sup of errors at the evaluation step and the sup of errors at the greedy step in ∞-norm. We thus recover the bounds computed by Patek (1997).

4. Application

The analysis of error propagation presented in Sect. 3.2 is general enough to develop several implementations. From the moment one can control the error made at each iteration step, the bound presented in Theorem 1 applies.

4.1. Algorithm

In this section, we present the Approximate Generalized Policy Iteration-Q (AGPI-Q) algorithm which is an extension for SG of Fitted-Q. This algorithm is offline and uses the so-called state-action value function Q. The state-action value function extends the value function by adding two degrees of freedom for the first action of each player. More formally, the state-action value function $Q^{\mu,\nu}(s,a,b)$ is defined as

$$Q^{\mu,\nu}(s,a,b) = E[r(s,a,b)] + \sum_{s'\in S} p(s'|s,a,b)v_{\mu,\nu}(s').$$

We assume we are given some samples $((x^j,a_1^j,a_2^j),r^j,x'^j)_{j=1,\ldots,N}$ and an initial Q-function (here we chose the null function). As it is an instance of AGPI, each iteration of this algorithm is made of a greedy step and an estimation step. The algorithm is precisely described in Algorithm 1.

Algorithm 1 AGPI - Q for Batch sample

Input: $((x^j,a_1^j,a_2^j),r^j,x'^j)_{j=1,\ldots,N}$ some samples,
$q_0=0$ a Q-function,
$\mathcal{F}$ an hypothesis space
for k=1,2,...,K **do**
  Greedy step:
  for all j **do**
    $\bar{a_j} = \arg\max_{\bar{a}} \min_{\bar{b}} q_{k-1}(x'^j,\bar{a},\bar{b})$ (solving a matrix game)
  end for
  Evaluation step:
  $q_{k,0}=q_{k-1}$
  for i=1,...,m **do**
    for all j **do**
      $q^j = r(x^j,a_1^j,a_2^j) + \gamma \min_b q_{k,i-1}(x'^j,\bar{a_j},b)$
    end for
    $q_{k,i} = \arg\min_{q\in\mathcal{F}} \sum_{j=1}^{N} l(q(x^j,a_1^j,a_2^j),q^j)$
    Where l is a loss function.
    $q_k = q_{k,m}$
  end for
end for
output q_K

For the *greedy step*, the minimax policy for the maximizer on each matrix game defined by $(q_k(x'^j,a,b))_{a,b}$. In general, this step involves solving N linear programs; recall

that in the case of a turn-based game this step reduces to finding a maximum. The *evaluation step* involves solving the MDP with an horizon m for the minimizer. This part is similar to fitted-Q iteration. At each step, we try to find the best fit over our hypothesis space for the next Q-function according to some loss function $l(x, y)$ (often, $l(x, y) = |x - y|^2$).

4.2. Analysis

For this algorithm, we have $\epsilon'_k = 0$ and ϵ_k the error made on q_k at each iteration. Let us note $\epsilon_{k,i}$ the error or fitting the Q-function on the feature space. The Bellman operator in the case of actions value function $Q(s, a_1, a_2)$ for policy μ and ν is $(\mathcal{T}_{\mu,\nu}\ Q)(s, a_1, a_2) = r(s, a_1, a_2) + \gamma \mathcal{P}_{\mu,\nu}\ (Q(., \mu(.), \nu(.)))$. The other non-linear operators are analogous to those defined in Sec. 2.2. In this section the operator used is the one on Q-function.

We have $q_{k,i+1} = \mathcal{T}_{\mu_k}\ q_{k,i} + \epsilon_i$. Let us define $\nu_{k,i}$ such as $\mathcal{T}_{\mu_k}{}^m q_{k,0} = \mathcal{T}_{\mu_k,\nu_{k,m-1}} ...\mathcal{T}_{\mu_k,\nu_{k,0}}\ q_{k,0}$. Furthermore we have $q_{k,i+1} \leq \mathcal{T}_{\mu_k,\nu_{k,i}}\ q_{k,i} + \epsilon_{k,i}$. On the one hand, we have:

$$\begin{aligned}
\epsilon_k &= q_{k,m} - \mathcal{T}_{\mu_k}{}^m q_{k,0}, \\
&\leq \mathcal{T}_{\mu_k,\nu_{k,m-1}} ...\mathcal{T}_{\mu_k,\nu_{k,0}}\ q_{k,0} \\
&\quad + \sum_{i=0}^{m-1} \mathcal{P}_{\mu_k,\nu_{k,m-1}} ...\mathcal{P}_{\mu_k,\nu_{k,i+1}}\ \epsilon_{k,i} \\
&\quad - \mathcal{T}_{\mu_k,\nu_{k,m-1}} ...\mathcal{T}_{\mu_k,\nu_{k,0}}\ q_{k,0}, \\
&\leq \sum_{i=0}^{m-1} \mathcal{P}_{\mu_k,\nu_{k,m-1}} ...\mathcal{P}_{\mu_k,\nu_{k,i+1}}\ \epsilon_{k,i}. \qquad (9)
\end{aligned}$$

On the other hand (with $\tilde{\nu}_{k,i}$ as $q_{k,i+1} = \mathcal{T}_{\mu_k,\tilde{\nu}_{k,i}}\ q_{k,i} + \epsilon_{k,i}$), we have:

$$\begin{aligned}
\epsilon_k &= q_{k,m} - \mathcal{T}_{\mu_k}{}^m q_{k,0}, \\
&\geq \mathcal{T}_{\mu_k,\tilde{\nu}_{k,m-1}} ...\mathcal{T}_{\mu_k,\tilde{\nu}_{k,0}}\ q_{k,0} \\
&\quad + \sum_{i=0}^{m-1} \mathcal{P}_{\mu_k,\tilde{\nu}_{k,m-1}} ...\mathcal{P}_{\mu_k,\tilde{\nu}_{k,i+1}}\ \epsilon_{k,i} \\
&\quad - \mathcal{T}_{\mu_k,\tilde{\nu}_{k,m-1}} ...\mathcal{T}_{\mu_k,\tilde{\nu}_{k,0}}\ q_{k,0}, \\
&\geq \sum_{i=0}^{m-1} \mathcal{P}_{\mu_k,\tilde{\nu}_{k,m-1}} ...\mathcal{P}_{\mu_k,\tilde{\nu}_{k,i+1}}\ \epsilon_{k,i}. \qquad (10)
\end{aligned}$$

From these inequalities, we can provide the following bound (the proof is given in Appendix B):

$$\begin{aligned}
\|l_k\|_{p,\rho} &\leq \frac{2(\gamma - \gamma^k)(1 - \gamma^m)}{(1 - \gamma)^3} (\mathcal{C}_q^{1,k,0,m,0})^{\frac{1}{p}} \sup_{i,l} \|\epsilon_{i,l}\|_{pq',\sigma} \\
&\quad + \frac{2\gamma^k}{1 - \gamma} (\mathcal{C}_q^{k,k+1,0})^{\frac{1}{p}} \min(\|d_0\|_{pq',\sigma}, \|b_0\|_{pq',\sigma}), \qquad (11)
\end{aligned}$$

with

$$\mathcal{C}_q^{l,k,l',k',d} = \frac{(1 - \gamma)^3}{(\gamma^l - \gamma^k)(\gamma^{l'} - \gamma^{k'})} \sum_{i=l}^{k-1} \sum_{i'=l'}^{k'-1} \sum_{j=i+i'}^{\infty} \gamma^j c_q(j + d).$$

4.3. Complexity analysis

At the *greedy step*, the algorithm solves N minimax equilibria for a zero-sum matrix game. This is usually done by linear programming. For state s, the complexity of such an operation is the complexity of solving a linear program with $c_s = 1 + card(A^1(s)) + card(A^2(s))$ constrains and with $card(A^1(s))$ variables. Let us note $\mathcal{L}(c_s, card(A^1(s)))$ this complexity. Then, the complexity of this step is bounded by $N\mathcal{L}(c, a)$ (with $c = \sup_{s \in \{x'^1,...,x'^N\}} c_s$ and $a = \sup_{s \in \{x'^1,...,x'^N\}} card(A^1(s))$). Using the simplex method, $\mathcal{L}(c, a)$ may grow exponentially with c while with the interior point method, $\mathcal{L}(c, a)$ is $O(a^{3.5})$ (Karmarkar, 1984). The time to compute q_j in the *evaluation step* depends on finding a maximum over $A^2(x'^j)$. And the regression complexity to find $q_{k,i}$ depends on the regression technique. Let us note this complexity $\mathcal{R}(N)$. Finally, the complexity of this step is $m\mathcal{R}(N)$.

The overall complexity is thus $O(N\mathcal{L}(c, a) + m\mathcal{R}(N))$; in general, the complexity of solving the linear program will be the limiting factor.

5. Experiments

In this section, AGPI-Q is tested on the Alesia game described in Sec. 2.5 where we assume that both players start with a budget $n = 20$. As a baseline, we use the exact solution of the problem provided by VI. We have run the algorithm for $K = 10$ iterations and for $m \in \{1, 2, 3, 4, 5\}$ evaluation steps. We have considered different sample set sizes, $N = 2500, 5000, 10000$. Each experiment is repeated 20 times. First, we generate N uniform samples (x^j) over the state space. Then, for each state, we draw uniformly the actions of each player in the set of their own action space in that state $a_1^j \sim \mathcal{U}(A^1(x^j))$, $a_2^j \sim \mathcal{U}(A^2(x^j))$, $r^j = r(x^j, a_1^j, a_2^j)$ and compute the next state x'^j. As hypothesis space, we use CART trees (Breiman et al., 1984) which exemplifies the non-parametric property of the algorithm.

The performance of the algorithm is measured as the mean-squared error between the value function $v_K(s) = \min_{\overline{b}} \max_{\overline{a}} q_K(s, \overline{a}, \overline{b})$ where q_K is the output of the algorithm AGPI-Q and the actual value function computed via VI. Figure 1 shows the evolution of performance along iterations for $N = 10000$ for the different values of the parameter m. Figure 2 shows the exact value function (Fig. 2(a)) and the approximated one v_K (Fig. 2(b)). The complete list of experiments results can be found in the supplementary

file, especially for different size of sample set $N = 2500$ and $N = 5000$.

For each size of sample set, the asymptotic convergence is better for small values of m. This is conform to Eq. (11), in which the term $\frac{2(\gamma-\gamma^k)(1-\gamma^m)}{(1-\gamma)^3}$ increases with m. However, for small values of k, the mean-squared error is reducing when m is increasing. This is coherent with experimental results when using MPI for MDP: the bigger m, the higher the convergence rate. The price to pay for this acceleration of convergence towards the optimal value is an heavier evaluation step. This is similar to results in the exact case (Puterman, 1994). Overall, this suggests to use large values of m at the beginning of the algorithm and to reduce it as k grows to get a smaller asymptotic error.

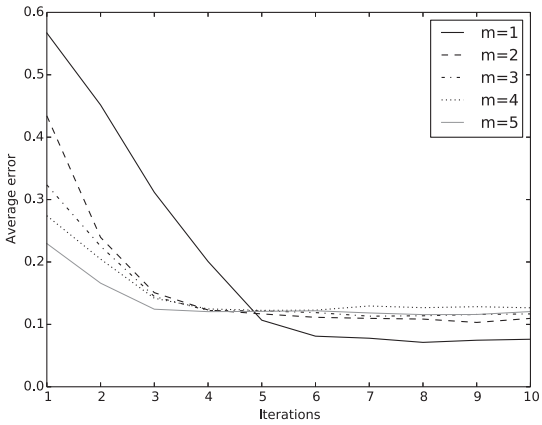
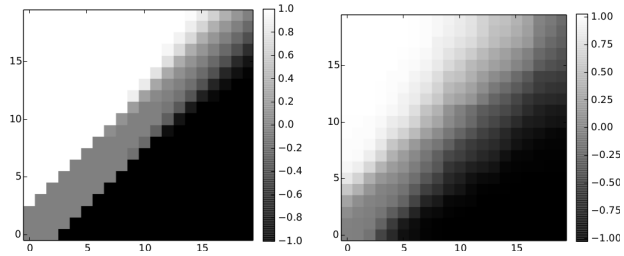


Figure 1. Mean-squared error (y-axis) between the estimated value function and the true value function at step k (x-axis). For $n = 20$ and $N = 10000$



(a) Exact value for the Alesia game for player's budget 0...20 and for token position 3

(b) Value computed by AGPI-Q for the game of Alesia for player's budget 0...20 and for token position 3

Figure 2. Value functions at token position 3

6. Conclusion and Perspectives

This work provides a novel and unified error propagation analysis in L_p-norm of well-known algorithms (API, AVI and AGPI) for zero-sum two-player SGs. It extends the error propagation analyses of Scherrer et al. (2012) for MDPs to zero-sum two-player SGs and of Patek (1997) which is an L_∞-norm analysis for only API. In addition, we provide a practical algorithm (AGPI-Q) which learns a good approximation of the Nash Equilibrium from batch data provided in the form of transitions sampled from actual games (the dynamics is not known). This algorithm is an extension of Fitted-Q for zero-sum two-player SGs and can thus be non-parametric. No features need to be provided or hand-crafted for each different application which is a significant advantage. Finally, we empirically demonstrate that AGPI-Q performs well on a simultaneous two-player game, namely Alesia.

It appears that the provided bound is highly sensitive to γ (which is a common problem of ADP). This is critical and further work should concentrate on reducing the impact of γ in the final error bound. Since non-stationary policies can reduce the impact of γ in MDP (Scherrer & Lesner, 2012), extensions of this work to zero-sum two-player SGs is forecasted. Moreover, we intend to apply AGPI-Q to large scale games and implement it on real data.

References

Antos, A., Szepesvári, C., and Munos, R. Fitted-Q Iteration in Continuous Action-Space MDPs. In *Proc. of NIPS*, pp. 9–16, 2008.

Bertsekas, D. P. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific Belmont, MA, 1995.

Bowling, M. and Veloso, M. Rational and Convergent Learning in Stochastic Games. In *Proc. of IJCAI*, volume 17, pp. 1021–1026, 2001.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. *Classification and Regression Trees*. CRC press, 1984.

Busoniu, L., Babuska, R., and De Schutter, B. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(2):156–172, March 2008.

Ernst, D., Geurts, P., and Wehenkel, L. Tree-Based Batch Mode Reinforcement Learning. In *Journal of Machine Learning Research*, pp. 503–556, 2005.

Farahmand, A.-M., Szepesvári, C., and Munos, R. Error Propagation for Approximate Policy and Value Iteration. In *Proc. of NIPS*, pp. 568–576, 2010.

Gabillon, V., Lazaric, A., Ghavamzadeh, M., and Scherrer, B. Classification-Based Policy Iteration with a Critic. In *Proc. of ICML*, pp. 1049–1056, 2011.

Greenwald, A., Hall, K., and Serrano, R. Correlated Q-learning. In *Proc. of ICML*, volume 3, pp. 242–249, 2003.

Hansen, T. D., Miltersen, P. B., and Zwick, U. Strategy Iteration is Strongly Polynomial for 2-Player Turn-Based Stochastic Games with a Constant Discount Factor. *JACM*, 60(1):1, 2013.

Hu, J. and Wellman, M. P. Nash Q-Learning for General-Sum Stochastic Games. *JMLR*, 4:1039–1069, 2003.

Karmarkar, N. A New Polynomial-time Algorithm for Linear Programming. In *Proc. of ACM Symposium on Theory of Computing*, pp. 302–311, 1984.

Lagoudakis, M. G. and Parr, R. Least-squares policy iteration. *Journal of Machine Learning Research*, pp. 1107–1149, 2003a.

Lagoudakis, M. G. and Parr, R. Reinforcement Learning as Classification: Leveraging Modern Classifiers. In *Proc. of ICML*, volume 3, pp. 424–431, 2003b.

Lagoudakis, Michail G and Parr, Ronald. Value function approximation in zero-sum markov games. In *Proc. of UAI*, pp. 283–292, 2002.

Lazaric, A., Ghavamzadeh, M., Munos, R., et al. Analysis of a Classification-Based Policy Iteration Algorithm. In *Proc. of ICML*, pp. 607–614, 2010.

Littman, M. L. Markov Games as a Framework for Multi-Agent Reinforcement Learning. In *Proc. of ICML*, volume 94, pp. 157–163, 1994.

Meyer, Christophe, Ganascia, Jean-Gabriel, and Zucker, Jean-Daniel. Learning strategies in games by anticipation. In *IJCAI 97, August 23-29, 1997, 2 Volumes*, pp. 698–707, 1997.

Munos, R. and Szepesvári, C. Finite-time bounds for fitted value iteration. *JMLR*, 9:815–857, 2008.

Patek, S. D. *Stochastic Shortest Path Games: Theory and Algorithms*. PhD thesis, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems, 1997.

Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

Scherrer, B. and Lesner, B. On the Use of Non-Stationary Policies for Stationary Infinite-Horizon Markov Decision Processes. In *Proc. of NIPS*, pp. 1826–1834, 2012.

Scherrer, B., Ghavamzadeh, M., Gabillon, V., and Geist, M. Approximate Modified Policy Iteration. In *Proc. of ICML*, 2012.

Shapley, L. S. Stochastic Games. *Proceedings of the National Academy of Sciences of the United States of America*, 39(10):1095, 1953.

Van Der Wal, J. Discounted Markov Games: Generalized Policy Iteration Method. *Journal of Optimization Theory and Applications*, 25(1):125–138, 1978.

Von Neumann, J. Morgenstern, 0.(1944) theory of games and economic behavior. *Princeton: Princeton UP*, 1947.