Policy Improvement for several Environments

Andreas Matt ANDREAS.MATT@UIBK.AC.AT
Georg Regensburger GEORG.REGENSBURGER@UIBK.AC.AT
Institute of Mathematics[1], University of Innsbruck, Austria

Abstract

In this paper we state a generalized form of the policy improvement algorithm for reinforcement learning. This new algorithm can be used to find stochastic policies that optimize single-agent behavior for several environments and reinforcement functions simultaneously. We first introduce a geometric interpretation of policy improvement, define a framework to apply one policy to several environments, and propose the notion of balanced policies. Finally we explain the algorithm and present examples.

1. Idea

Until now reinforcement learning has been applied to learn behavior within one environment. Several methods to find optimal policies for one environment are known (Kaelbling et al., 1996; Sutton & Barto, 1998).

In our research we focus on a general point of view of behavior that appears independently from a single environment. As an example imagine that a robot should learn to avoid obstacles, a behavior suitable for more than one environment. Obviously a policy for several environments cannot - in general - be optimal for each one of them. Improving a policy for one environment may result in a worse performance in an other. Nevertheless it is often possible to improve a policy for several environments. Compared to multiagent reinforcement learning as in Bowling and Veloso (2000), where several agents act in one environment, we have one agent acting in several environments.

2. Equivalent and Improving Policies

We fix a finite Markov Decision Process $(S, \mathbf{A}, \mathbf{P}, \mathbf{R})$, use the standard definitions of value function V and Q-value and write $\pi(a \mid s)$ for the probability that action a is chosen in state s. We say that two policies π and $\tilde{\pi}$ are equivalent if their value functions coincide, i.e. $V^\pi = V^{\tilde{\pi}}$.

Theorem 1 *Two policies $\tilde{\pi}$ and π are equivalent if and only if*

$$\sum\nolimits_{a\in A} Q^\pi(a,s)\tilde{\pi}(a \mid s) = V^\pi(s) \text{ for all } s \in S.$$

This gives us a description of the equivalence class of a policy. We interpret $\pi(- \mid s)$ as a point on a standard simplex and the equivalence class as the intersection of the hyperplane H defined by $Q^\pi(- \mid s)$ and $V^\pi(s)$ with the simplex. See Figure 1 *left* for an example with three actions a_1, a_2 and a_3. The following theorem is

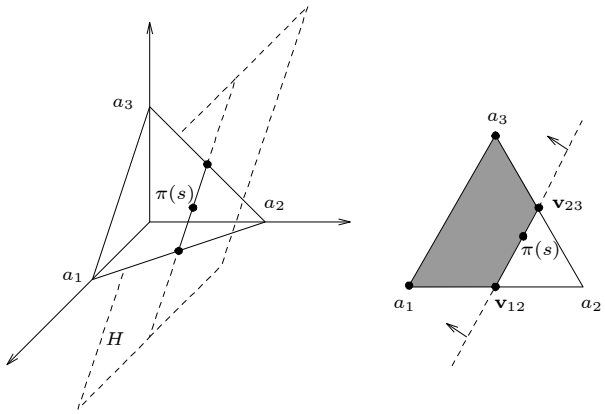

Figure 1. left: A policy in state s and its equivalence class *right:* Improving policies in state s

a general version of the policy improvement theorem.

Theorem 2 *Let π and $\tilde{\pi}$ be policies such that*

$$\sum\nolimits_{a\in A} Q^\pi(a,s)\tilde{\pi}(a \mid s) \geq V^\pi(s) \text{ for all } s \in S.$$

Then $V^{\tilde{\pi}} \geq V^\pi$. If additionally there exists an $s \in S$ such that $\sum Q^\pi(a,s)\tilde{\pi}(a \mid s) > V^\pi(s)$ then $V^{\tilde{\pi}} > V^\pi$.

We define the *set of improving policies* for π in s by

$$C^\pi_{\geq}(s) = \left\{\tilde{\pi}(- \mid s) : \sum Q^\pi(a,s)\tilde{\pi}(a \mid s) \geq V^\pi(s)\right\},$$

[1]We wish to thank Prof. Ulrich Oberst for his motivation, comments and support. This research was partially supported by "Forschungsstipendien an österreichische Graduierte" and Project Y-123 INF.

and in analogy the *set of strictly improving policies* $C^\pi_>(s)$ and the *set of equivalent policies* $C^\pi_=(s)$ for π in s. We define the *set of strictly improving actions* of π in s by $A^\pi_>(s) = \{a : Q^\pi(a,s) > V^\pi(s)\}$.

The set of improving policies $C^\pi_\geq(s)$ is a polytope given by the intersection of a half-space and a standard simplex. Its vertices are $\text{vert}\left(C^\pi_\geq(s)\right) = \text{vert}(C^\pi_=(s)) \cup A^\pi_>(s)$. See Figure 1 *right*, where $A^\pi_>(s) = \{a_1, a_3\}$, $\text{vert}(C^\pi_=(s)) = \{\mathbf{v}_{12}, \mathbf{v}_{23}\}$ and $C^\pi_\geq(s)$ is the shaded area, the side marked by the small arrows.

3. Policies for several Environments

Consider a robot and its sensors to perceive the world. All possible sensor values together represent all possible states for the robot. In each of these states the robot can perform some actions. We call all possible states and actions the *state action space (SAS)* $\mathbb{E} = (S, \mathbf{A})$. Now we put the robot in a physical environment, where we can observe all possible states for this environment, a subset $S_E \subset S$ of all possible states in general, and the transition probabilities $\mathbf{P}_E$. We call $E = (S_E, \mathbf{A}, \mathbf{P}_E)$ a *realization* of an SAS.

Let $\mathbf{E} = (E_i, \mathbf{R}_i)_{i=1\ldots n}$ be a finite family of realizations of an SAS with rewards $\mathbf{R}_i$. Since the actions are given by the SAS it is clear what is meant by a policy π for $\mathbf{E}$. For each $(E_i, \mathbf{R}_i)$ we can calculate the value function, which we denote by V_i^π. We define the *set of improving policies* of π in $s \in S$ by

$$C^\pi_\geq(s) = \bigcap\nolimits_{i\in[n],\ s\in S_i} C^\pi_{i,\geq}(s)$$

and the *set of strictly improving policies* of π in s by

$$C^\pi_>(s) = \left\{ \begin{array}{c} \tilde\pi(- \mid s) \in C^\pi_\geq(s) \text{ such that} \\ \exists\, i \in [n] \text{ with } \tilde\pi(- \mid s) \in C^\pi_{i,>}(s) \end{array} \right\},$$

where $[n] = \{1, \ldots, n\}$. The set of improving policies of π in s is the intersection of a finite number of half-spaces through a point with a standard simplex.

Theorem 3 *Let $\tilde\pi$ be a policy for $\mathbf{E}$ such that*

$$\tilde\pi(- \mid s) \in C^\pi_\geq(s) \text{ for all } s \in S.$$

Then $V_i^{\tilde\pi} \geq V_i^\pi$ for all $i \in [n]$. If additionally there exist an $s \in S$ with $\tilde\pi(- \mid s) \in C^\pi_>(s)$ then there exists an $i \in [n]$ such that $V_i^{\tilde\pi} > V_i^\pi$.

In order to describe $C^\pi_\geq(s)$ and find an $\tilde\pi(- \mid s) \in C^\pi_>(s)$ we consider its vertices. We call the vertices of $C^\pi_\geq(s)$ *improving vertices* and define the *strictly improving vertices* by $\text{vert}(C^\pi_>(s)) = \text{vert}(C^\pi_\geq(s)) \cap C^\pi_>(s)$. There exist several algorithm to find all vertices of a polytope (Fukuda, 2000). Linear Programming methods can be used to decide whether there exist strictly improving vertices and to find one (Schrijver, 1986). Observe that for a single environment the strictly improving vertices are just the set of strictly improving actions.

Let $s \in S$. We define π_s as the set of all policies that are arbitrary in s and equal π otherwise. We call a policy *balanced* if and only if for all $s \in S$ and all $\tilde\pi \in \pi_s$ either $V_i^{\tilde\pi} = V_i^\pi$ for all $i \in [n]$ or there exists $i \in [n]$ such that $V_i^{\tilde\pi} < V_i^\pi$. This means that if one changes a balanced policy in one state s it is the same for all environments or it gets worse in at least one. Compare to the notion of an equilibrium point in game theory (Nash, 1951). Note that for one environment the notions of optimal and balanced policies coincide.

Theorem 4 *A policy π is balanced if and only if there are no strictly improving policies, i.e. $C^\pi_>(s) = \emptyset$ for all $s \in S$.*

4. General Policy Improvement

We state a generalized form of the policy improvement algorithm for a family of realizations of an SAS which we call *general policy improvement* (algorithm 1). The idea is to improve the policy by choosing in each state a strictly improving vertex. If there are no strictly improving vertices the policy is balanced and the algorithm terminates.

Input: a policy π and a family of realizations $(E_i, \mathbf{R}_i)$
Output: a balanced policy $\tilde\pi : V_i^{\tilde\pi} \geq V_i^\pi$ for all $i \in [n]$

```
π̃ ⟵ π
repeat
    calculate V_i^π̃ and Q_i^π̃ for all i ∈ [n]
    for all s ∈ S do
        if vert(C_>^π̃(s)) ≠ ∅ then
            choose π'(− | s) ∈ vert(C_>^π̃(s))
            π̃(− | s) ⟵ π'(− | s)
until vert(C_>^π̃(s)) = ∅ for all s ∈ S
```

Algorithm 1: General Policy Improvement

In each step of the algorithm we try to choose a strictly improving vertex. Different choices may result in different balanced policies and influence the number of improvement steps before termination. The algorithm includes policy improvement for one environment as a special case.

A geometric interpretation of one step of the general policy improvement algorithm for three states and two realizations can be seen in Figure 2. In state s_1 there are no strictly improving vertices. In state s_2 there are three strictly improving vertices, one of them is the action a_3. In state s_3 there are only two, both of

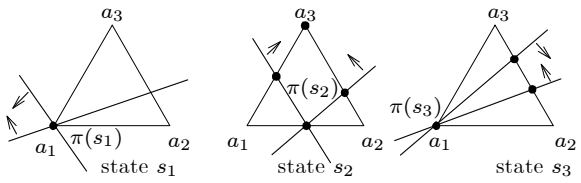

Figure 2. Improving policies for two realizations

them a stochastic combination of a_2 and a_3.

5. Examples

All experiments are made with a 10x10 gridworld simulator to learn an obstacle avoidance behavior. The robot has 4 sensors, forward, left, right, and back, with a range of 5 blocks each. There are 3 actions in each state: move forward, left and right. The robot gets rewarded if it moves away from obstacles, it gets punished if it moves towards obstacles.

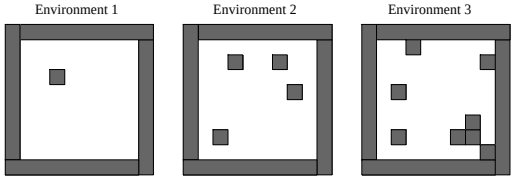

Figure 3. Three different environments

We choose three environments (see Figure 3) with the same reinforcement function and run the algorithm. In all experiments we start with the random policy. We calculate in each step all strictly improving vertices and choose one randomly. In order to evaluate and compare policies we consider the average utilities of all states, and normalize it, with 1 being an optimal and 0 the random policy in this environment. Four sample experiments show performances in each environment of the different balanced policies learned.

Experiment:	1	2	3	4
Environment 1	0.994	0.771	0.993	0.826
Environment 2	0.862	0.876	0.788	0.825
Environment 3	0.872	0.905	0.975	0.878

Figure 4 shows the progress of the algorithm for each environment in experiment 2. In all experiments the algorithm terminates after 6 to 8 iteration steps.

6. Discussion

The general policy improvement algorithm can be used to improve a policy for several realizations of a state

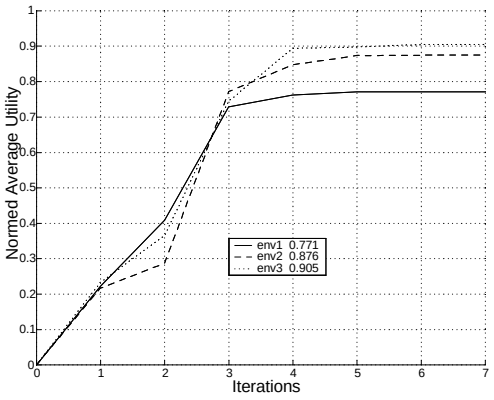

Figure 4. The progress of general policy improvement for each environment.

action space simultaneously. This means that it can be used to learn a policy for several environments and several reinforcement functions together. A useful application is to add a new environment or behavior to an already optimal policy, without changing its performance. We have already implemented Value Iteration for several realizations which leads to an extension of Q-learning. Our future research focuses on the implementation of on-line algorithms, methods to find the strictly improving vertices and to decide which of them are best regarding to learning speed. For more detailed information please consult the extended version of this paper on `http://mathematik.uibk.ac.at/~rl`.

References

Bowling, M., & Veloso, M. (2000). *An analysis of stochastic game theory for multiagent reinforcement learning* (Technical Report CMU-CS-00-165).

Fukuda, K. (2000). Frequently asked questions in polyhedral computation. `http://www.ifor.math.ethz.ch/~fukuda/polyfaq/polyfaq.html`.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, pp. 237–285.

Nash, J. (1951). Non-cooperative games. *Ann. of Math. (2)*, *54*, 286–295.

Schrijver, A. (1986). *Theory of linear and integer programming*. Chichester: John Wiley & Sons Ltd.

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