Approximate Policy Iteration for several Environments and Reinforcement Functions

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

Abstract

We state an approximate policy iteration algorithm to find stochastic policies that optimize single-agent behavior for several environments and reinforcement functions simultaneously. After introducing a geometric interpretation of policy improvement for stochastic policies we discuss approximate policy iteration and evaluation. We present examples for two blockworld environments and reinforcement functions.

1. Introduction

Reinforcement learning methods usually achieve optimal policies for one reinforcement function in one environment (Bertsekas & Tsitsiklis, 1996; Kaelbling et al., 1996). Multicriteria reinforcement learning is concerned with optimizing several reinforcement functions in one environment. Gábor et al. (1998) order the different criteria, Wakuta (1995) discusses policy improvement to find optimal deterministic policies for vector-valued Markov decision processes. In our research we focus on finding stochastic policies, which can perform better than deterministic policies, for several environments and reinforcement functions.

2. MDP's and State Action Space

An *environment* $E = (S, \mathbf{A}, \mathbf{P})$ is given by a finite set S of *states*, a family $\mathbf{A} = (A(s))_{s\in S}$ of finite sets of *actions* and a family $\mathbf{P} = P(- \mid a, s)_{s\in S, a\in A(s)}$ of *transition probabilities* on S. A *policy* for E is given by a family $\pi = \pi(- \mid s)_{s\in S}$ of probabilities on $A(s)$. A *Markov decision process (MDP)* is given by an environment $E = (S, \mathbf{A}, \mathbf{P})$ and a family $\mathbf{R} = R(s', a, s)_{s',s\in S, a\in A(s)}$ of *rewards* in $\mathbb{R}$. Let $(E, \mathbf{R})$ be a MDP and $0 \leq \gamma < 1$ be a discount rate. Let π be a policy for E. We denote by $V^\pi(s)$ the (discounted) *value function* or *utility* of policy π in state s and write $Q^\pi(a, s)$ for the (discounted) *action-value*. The goal of reinforcement learning is to find policies that maximize the value function.

To apply one policy to different environments we introduce the following notions. A *state action space* $\mathbb{E} = (S, \mathbf{A})$ is given by a finite set S of states and a family $\mathbf{A} = (A(s))_{s\in S}$ of finite sets of actions. We call an environment $E = (S_E, \mathbf{A}_E, \mathbf{P}_E)$ and a MDP $(E, \mathbf{R})$ a *realization* of a state action space if the set of states is a subset of S and the actions for all states are the same as in the state action space, that is $S_E \subset S$ and $\mathbf{A}_E(s) = \mathbf{A}(s)$ for all $s \in S$. We can define policies for a state action space which can be applied to any realization.

3. Policy Improvement

3.1 One Realization

The policy improvement theorem for stochastic policies gives a sufficient criterion to improve a given policy π. Let $\tilde{\pi}$ be a policy such that

$$\sum\nolimits_{a\in A} Q^\pi(a, s)\tilde{\pi}(a \mid s) \geq V^\pi(s) \text{ for all } s \in S. \quad (1)$$

Then $V^{\tilde{\pi}} \geq V^\pi$, that is $V^{\tilde{\pi}}(s) \geq V^\pi(s)$ for all $s \in S$. If additionally there exists an $s \in S$ such that the inequality (1) is strict then $V^{\tilde{\pi}} > V^\pi$. A usual choice for *policy improvement* is $\tilde{\pi}(a \mid s) = 1$ for an action $a \in A(s)$ such that $Q^\pi(a, s) = \max_a Q^\pi(a, s)$. Repeating policy improvement leads to *policy iteration*.

Considering all stochastic policies satisfying (1), we define the *set of improving policies* for π in s by

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

The *set of strictly improving policies* $C^\pi_{>}(s)$ and the *set of equivalent policies* $C^\pi_{=}(s)$ for π in s are defined analogously. 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

$$\operatorname{vert}\left(C^\pi_{\geq}(s)\right) = \operatorname{vert}(C^\pi_{=}(s)) \cup A^\pi_{>}(s). \quad (2)$$

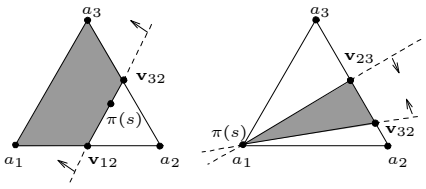

Figure 1. left: Improving policies for one realization *right:* Improving policies for two realizations

See Figure 1 *left*, where $C^\pi_{\geq}(s)$ is the shaded area, the side marked by the small arrows, $A^\pi_{>}(s) = \{a_1, a_3\}$ and $\text{vert}(C^\pi_{=}(s)) = \{\mathbf{v}_{12}, \mathbf{v}_{32}\}$. Let $A(s) = \{a_1, \ldots, a_n\}$, $\mathbf{c} = (c_1, \ldots, c_n) \in \mathbb{R}^n$ with $c_i = Q^\pi(a_i, s)$ and $b = V^\pi(s)$. Let $\mathbf{e}_i \in \mathbb{R}^n$ denote the ith standard basis vector. Then $\text{vert}\,(C^\pi_{=}(s))$ is $\{\mathbf{e}_k : c_k = b\} \cup$

$$\left\{\mathbf{v}_{ij} = \frac{b - c_j}{c_i - c_j}\mathbf{e}_i + \frac{c_i - b}{c_i - c_j}\mathbf{e}_j : c_i > b,\, c_j < b\right\}. \quad (3)$$

3.2 Several Realizations

Let $\mathbb{E} = (S, \mathbf{A})$ be a state action space. We want to improve a policy π for two realizations $(E_1, \mathbf{R}_1)$ and $(E_2, \mathbf{R}_2)$ of $\mathbb{E}$ with value functions V^π_i and Q^π_i for discount rates γ_i, $i = 1, 2$. Let $\tilde{\pi}$ be a policy such that

$$\sum\nolimits_{a \in A} Q^\pi_1(a, s)\tilde{\pi}(a \mid s) \geq V^\pi_1(s) \text{ and} \quad (4)$$

$$\sum\nolimits_{a \in A} Q^\pi_2(a, s)\tilde{\pi}(a \mid s) \geq V^\pi_2(s) \quad (5)$$

for all $s \in S_1 \cap S_2$ and that $\tilde{\pi}\,(- \mid s)$ satisfies (4) or (5) if s is only contained in S_1 or S_2 respectively. Then $V^{\tilde{\pi}}_1 \geq V^\pi_1$ and $V^{\tilde{\pi}}_2 \geq V^\pi_2$, see equation (1).

We define the *set of improving policies* $C^\pi_{\geq}(s) = C^\pi_{1,\geq}(s) \cap C^\pi_{2,\geq}(s)$ for π in $s \in S_1 \cap S_2$. The *set of strictly improving policies* $C^\pi_{>}(s)$ is given by all policies $\tilde{\pi}\,(- \mid s) \in C^\pi_{\geq}(s)$ such that one inequality (4) or (5) is strict. If s is only contained in S_1 or S_2 we use the definition from the previous subsection. Let $\tilde{\pi}$ a policy such that $\tilde{\pi}(- \mid s) \in C^\pi_{\geq}(s)$ for all $s \in S$ and $\tilde{\pi}(- \mid s) \in C^\pi_{>}(s)$ for at least one s. Then $\tilde{\pi}$ performs better than π since $V^{\tilde{\pi}}_i \geq V^\pi_i$ and $V^{\tilde{\pi}}_1 > V^\pi_1$ or $V^{\tilde{\pi}}_2 > V^\pi_2$ by the previous subsection.

We call a policy *balanced* if $C^\pi_{>}(s)$ is empty. In general there exist several balanced policies which can be stochastic. In one environment a policy is optimal if and only if it is balanced. For further details and policy iteration for the general case with a finite family of realizations see Matt and Regensburger (2001).

To describe $C^\pi_{\geq}(s)$ and find a $\tilde{\pi}(- \mid s) \in C^\pi_{>}(s)$ we consider its vertices, see Figure 1 *right*. We call the vertices $\text{vert}(C^\pi_{\geq}(s))$ *improving vertices* and define the *strictly improving vertices* by $\text{vert}(C^\pi_{>}(s)) = \text{vert}(C^\pi_{\geq}(s)) \cap C^\pi_{>}(s)$. For one realization the strictly improving vertices are the strictly improving actions. To find all strictly improving vertices for two realizations and an $s \in S_1 \cap S_2$ we take all elements from $\text{vert}(C^\pi_{1,\geq}(s)) \cup \text{vert}(C^\pi_{2,\geq}(s))$ that are in $C^\pi_{1,>}(s)$ or $C^\pi_{2,>}(s)$, where $\text{vert}(C^\pi_{i,\geq}(s))$ are given by (2) and (3).

4. Approximate Policy Iteration

For policy iteration we need the action-values. If the model of the environment is not given explicitly we can approximate them. We use a SARSA related method, see algorithm 1, (Sutton & Barto, 1998).

repeat
  choose $s \in S$ and $a \in A(s)$ derived from π
  take action a and observe r and s'
  choose $a' \in A(s')$ according to π
  $Q(a, s) \longleftarrow Q(a, s) + \alpha\,(r + \gamma Q(a', s') - Q(a, s))$

Algorithm 1: Approximate Policy Evaluation

We call Algorithm 2 *approximate policy iteration* for several realizations. We start with an arbitrary policy and approximate the action-values. The value function can be derived by the *Bellman equation*. Then we improve the policy according to Section 3.2 with the approximated values.

repeat
  approximate V^π_i and Q^π_i for $i \in [n]$
  for all $s \in S$ **do**
    if $\text{vert}(C^{\tilde{\pi}}_{>}(s)) \neq \emptyset$ **then**
      choose $\pi'(- \mid s) \in \text{vert}(C^\pi_{>}(s))$
      $\pi(- \mid s) \longleftarrow \pi'(- \mid s)$

Algorithm 2: Approximate Policy Iteration

5. Experiments

All experiments are made with the simulator SimRobo[1]. The robot has four sensors, forward, left, right, and back, with a range of five blocks each. There are three actions in each state, move forward, left and right. The state action space is defined by all possible sensor values and actions. We consider two environments, see Figure 2, and two reinforcement functions. For *obstacle avoidance*, $\mathbf{R}_{oa}$, the robot gets rewarded if it moves away from obstacles and it gets punished if it moves towards them. For *wall following*, $\mathbf{R}_{wf}$, the robot gets rewarded if there is a block on its right side and punished otherwise.

[1] More information on the blockworld simulator is available at `http://mathematik.uibk.ac.at/users/rl`.

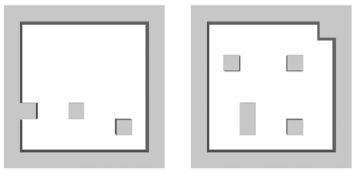

Figure 2. Environments E_1 and E_2

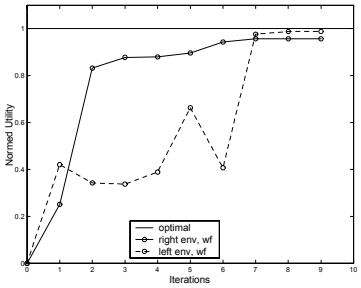

Figure 3. The performance of online policy iteration for wall following in two environments

In each experiment we consider two realizations and run Algorithm 2 starting with the random policy. For policy evaluation, Algorithm 1, we use discount rate $\gamma = 0.95$, start with learning rate $\alpha = 0.8$ and run 5000 iterations. We choose action a using the $\epsilon-$greedy policy derived from π. We define the *utility of a policy* by the average utilities of all states. To evaluate and compare the policies obtained after each improvement step we calculate the utilities of the policies exactly using value iteration. The utilities are then normalized, with 1 being an optimal and 0 the random policy in this realization.

5.1 Two Environments

We want to learn a policy for a wall following behavior for the environments $E_1 = (S_1, \mathbf{A}_1, \mathbf{P}_1)$ and $E_2 = (S_2, \mathbf{A}_2, \mathbf{P}_2)$. Thus we have the realizations $(E_1, \mathbf{R}_{wf})$ and $(E_2, \mathbf{R}_{wf})$. Figure 3 shows the utilities of the learned policy in each iteration step for each realization. Since the action values and value functions are only approximated the utility may decrease after a policy improvement step.

5.2 Two Environments and Two Reinforcement Functions

We look for a policy that avoids obstacles in E_1 and follows the wall in E_2. The realizations are $(E_1, \mathbf{R}_{oa})$ and $(E_2, \mathbf{R}_{wf})$. Figure 4 shows the utilities. Even tough wall following and obstacle avoidance together may be contradicting we obtain a stochastic policy that performs well in both realizations.

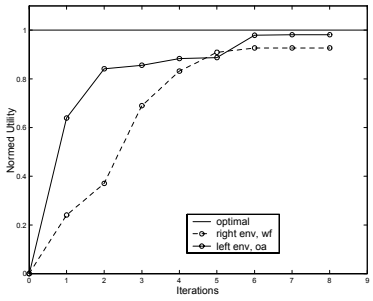

Figure 4. The performance of online policy iteration for wall following and obstacle avoidance in two environments

6. Discussion

Approximate policy iteration requires good approximations of all action-values in all realizations for the improvement step. Therefore the approximate policy evaluation step is critical and exploration plays a fundamental role. We note that the starting policy influences the policy learned by the algorithm. Our future research focuses on optimistic policy iteration methods, where the policy is improved after incomplete evaluation steps.

References

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

Gábor, Z., Kalmár, Z., & Szepesvári, C. (1998). Multi-criteria reinforcement learning. *Proceedings of the 15th International Conference on Machine Learning (ICML 1998)* (pp. 197–205). Madison, WI: Morgan Kaufmann.

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

Matt, A., & Regensburger, G. (2001). Policy improvement for several environments. *Proceedings of the 5th European Workshop on Reinforcement Learning (EWRL-5)* (pp. 30–32). Utrecht, Netherlands.

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

Wakuta, K. (1995). Vector-valued Markov decision processes and the systems of linear inequalities. *Stochastic Process. Appl., 56*, 159–169.