

Paper:

Opposition-Based Reinforcement Learning

Hamid R. Tizhoosh

Pattern Analysis and Machine Intelligence Laboratory, Systems Design Engineering, University of Waterloo
200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1

E-mail: tizhoosh@uwaterloo.ca

[Received September 29, 2005; accepted November 22, 2005]

Reinforcement learning is a machine intelligence scheme for learning in highly dynamic, probabilistic environments. By interaction with the environment, reinforcement agents learn optimal control policies, especially in the absence of a priori knowledge and/or a sufficiently large amount of training data. Despite its advantages, however, reinforcement learning suffers from a major drawback – high calculation cost because convergence to an optimal solution usually requires that all states be visited frequently to ensure that policy is reliable. This is not always possible, however, due to the complex, high-dimensional state space in many applications. This paper introduces opposition-based reinforcement learning, inspired by opposition-based learning, to speed up convergence. Considering opposite actions simultaneously enables individual states to be updated more than once shortening exploration and expediting convergence. Three versions of Q-learning algorithm will be given as examples. Experimental results for the grid world problem of different sizes demonstrate the superior performance of the proposed approach.

Keywords: reinforcement learning, Q-learning, opposite action, opposite state

1. Introduction

The machine intelligence field features many examples of ideas borrowed from nature, simplified, and modified for implementation in a numerical framework. Reinforcement learning [1, 6, 10] is one example of building on reward and punishment concepts central to human and animal learning. Dogs, for example, learn by rewards for desired responses. Reinforcement learning agents learn by accumulating knowledge (reinforcement signals) for selecting actions producing the highest rewards.

Reinforcement learning agents can autonomously explore highly dynamic, stochastic environments and develop optimal control policies, by accumulating evaluative feedback from the environment. This is specially desirable if a priori knowledge is not available and a large training set is absent. One problem arising with reinforcement learning algorithms, however, is the long time re-

quired for exploring the unknown environment.

Reinforcement learning convergence is generally guaranteed if all states can be visited infinitely [2, 8]. In practice, of course, computational resources are limited and application-based time constraints usually do not allow lengthy processing. Much reinforcement learning research has thus been devoted to state reduction or generalizing the experience of reinforcement learning agents. Reibero [7] argued that reinforcement agents are rarely a *tabula rasa* and intensive exploration may not be required, proposing the embedding of a priori knowledge about the environment. Combining reinforcement agents with other techniques is another approach, e.g. decision trees are used to reduce sample space [3]. Prior knowledge is also used to improve reinforcement learning agents [4]. Modified reinforcement agents such as relational reinforcement agents [5], have also been proposed.

Opposition-Based Learning [11] extends existing learning algorithms practically assuming simultaneous estimates and counter-estimates can be used to improve reinforcement agents and to shorten exploration time. In this work, a new class of reinforcement algorithms based on *opposition* will be introduced. Updating can be continuously repeated by considering states and opposite states and actions and opposite actions simultaneously shortening exploration period while maintaining the desired accuracy for the optimal action policy.

This paper is organized as follows: Section 2 introduces the opposition-based learning concept, while section 3 details opposition-based reinforcement learning. Section 4 applies the proposed approach to Q-learning introducing three possible extensions. Section 5 reviews the results of simulation and experiments, and section 6 presents conclusions.

2. Opposition-Based Learning

Learning, optimization, and search are basic to machine intelligence research. Algorithms learn from data or instructions, optimize estimated solutions, and search large spaces for solutions. Algorithms are inspired by diverse biological, behavioral, and natural phenomena.

When looking for solution x to a given problem, we usually make estimate \hat{x} , which is not an exact solution but based on experience or on a totally random guess.

Guesses usually involve complex problems, e.g. random initialization of weights in a neural net. In some cases, estimate \hat{x} is sufficient while in others, we seek increased accuracy in results. If the task of many intelligent techniques is understood as *function approximation*, we generally must cope with computational complexity. This means that although a solution may be reached, the required computation time may be beyond practical application – *the curse of dimensionality*.

Learning often begins randomly from scratch and moves toward solution. Weights in a neural network are randomly initialized, for example, the parameter population in genetic algorithms is configured randomly, and the action policy of reinforcement agents is initially based on randomness. The random guess, if near the optimal solution, may result in fast convergence. If the random guess is far from the existing solution, e.g. a worst-case situation, it is in the *opposite location*, so approximation, search or optimization requires considerably more time, or may not be attained. The absence of a priori knowledge may prevent the best initial guess from being made. Logically, we should look in all directions simultaneously, or, more concretely, in the opposite direction. As with *social revolutions*, which attempt to establish fundamental improvements by radical movement toward an opposite situation, searching in the opposite direction may be advantageous with algorithms.

If we are searching for solution x and we agree that searching in the opposite direction could be advantageous, the first step becomes calculating *opposite number* \check{x} [11].

Definition: Let $x \in R$ be a real number defined on a certain interval: $x \in [a, b]$. Opposite number \check{x} is defined as follows:

$$\check{x} = a + b - x. \quad \dots \dots \dots (1)$$

For $a = 0$ and $b = 1$, we receive

$$\check{x} = 1 - x. \quad \dots \dots \dots (2)$$

The opposite number in a multidimensional case is defined analogously.

Definition: Let $P(x_1, x_2, \dots, x_n)$ be a point in n -dimensional coordinates with $x_1, x_2, \dots, x_n \in R$ and $x_i \in [a_i, b_i] \quad \forall i \in \{1, 2, \dots, n\}$. Opposite point \check{P} is completely defined by its coordinates $\check{x}_1, \check{x}_2, \dots, \check{x}_n$, where

$$\check{x}_i = a_i + b_i - x_i. \quad \dots \dots \dots (3)$$

The opposition-based scheme for learning now becomes concrete [11]:

Opposition-Based Learning: Let $f(x)$ be the function in focus and $\hat{h}(\cdot)$ a proper evaluation function. If $x \in [a, b]$ is an initial (random) guess and \check{x} is its opposite, then in each iteration, we calculate $f(x)$ and $f(\check{x})$. Learning continues with x if $\hat{h}(f(x)) \geq \hat{h}(f(\check{x}))$, otherwise with \check{x} . As a measure of optimality, evaluation function $\hat{h}(\cdot)$ compares the suitability of results, e.g., fitness function, reward and punishment, and error function.

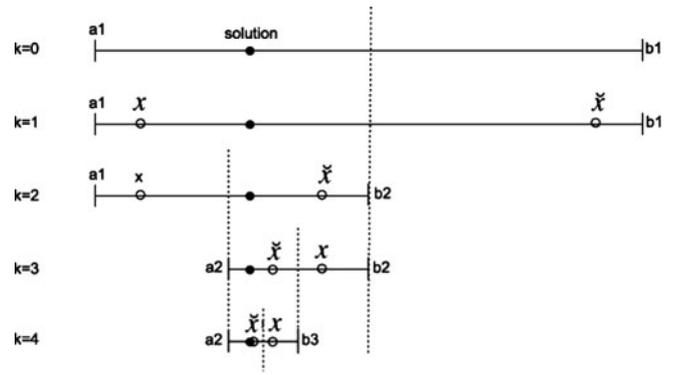


Fig. 1. Solving a one-dimensional equation via recursive halving of the search interval for optimizing estimates x and opposite estimates \check{x} .

Considering interval $[a_1, b_1]$ in **Fig.1**, the solution to a given problem is found by repeatedly examining guesses and counter-guesses. Opposite number \check{x} for initial guess x is generated. Based on whether the estimate or the counter-estimate is closer to the solution, the search interval is recursively halved until one of the two is close to an existing solution. The extension of this approach to a two-dimensional case is illustrated in **Fig.2**.

In [11] it was demonstrated that, based on opposition-based jumps from estimates to counter-estimates and vice versa, nonlinear problems are solved, improving the convergence of genetic algorithms, neural nets, and reinforcement agents. The experiments are however to the degree that extensive verification is still required.

The opposition-based learning concept is used in the sections that follow to introduce a new class of reinforcement agents.

3. Opposition-Based Reinforcement

Reinforcement learning is based on an intelligent agent interacting with the environment, and receiving rewards and punishment [10]. The agent acts to change the environment and is correspondingly rewarded or punished. Reinforcement learning is in this sense *weakly* supervised learning. To explain how opposition-based learning extends reinforcement agents, we focus on the simplest, most widely used reinforcement algorithm, Q-learning [13, 14]. General investigations on reinforcement agents based on temporal differences [9, 10] ($\lambda \neq 0$) remain the subject for projected work.

The time required for convergence in Q-learning is proportional to the size of the Q-matrix. A larger Q-matrix resulting from a larger number of states and/or a greater number of actions requires more time for it to be *filled*.

Reinforcement agents generally begin from scratch and make stochastic decisions, explore the environment, find rewarding actions, and exploit them. Initial performance of these agents is poor due to the lack of knowledge about