# Optimal Continuous Policies: A Classifier System Approach

Stewart W. Wilson

Prediction Dynamics, Concord MA 01742 USA
Department of General Engineering
The University of Illinois at Urbana-Champaign IL 61801 USA
wilson@prediction-dynamics.com

## Extended Abstract

The paper presents an extended classifier system that learns optimal continuous-valued actions in Markov environments where the state input $x$ is a real vector and the payoff function $R(x, a)$ is continuous. The system can be described as learning an optimal continuous policy $a^*(x)$ where $a^*$ is the highest paying action value, given $x$.

The system, shown schematically in Fig. 1, consists of two classifier systems S1 and S2 working in tandem.

The lower system S1 is in contact with an environment *Env* that provides real state vectors $x$ to S1, receives discrete actions $a_k$ from S1, and provides real payoffs $R$ to S1 in response. The system S1 is based on XCS-LP [1], a classifier system designed for environments with real state vectors and payoffs. For each possible $x$, S1 learns the payoff $R$ to be expected for each $a_k$ it might take.

While S1 is learning, the system S2 *observes* S1's *prediction array.* Thus S2 can see what payoff S1 expects for each $a_k$. In particular, S2 notes which $a_k$ has the *maximum* expected payoff, and uses this action value, termed $a_k^*$, as an input. The other input to S2 is $x$. S2's objective is to learn to predict, given $x$, the correct value of $a_k^*$.

The combined system S1-S2 has two innovations. First, once S2 has learned to predict correctly, it represents a *direct* mapping from $x$ to the optimal action, thus embodying the optimal policy $a^*(x)$. Ordinarily, RL systems do not directly map from $x$ to $a^*(x)$. Instead, as with S1, they learn expected payoffs for each $a_k$, then, given an $x$, choose the $a_k$ with the maximum expectation and send it to the environment. In contrast, S2 sees $x$ and predicts $a^*$ directly.

This somewhat academic difference becomes more interesting because the architecture permits $a^*(x)$ to be *continuous-valued*, despite the fact that S2 is only capable of discrete actions. Like S1, S2 is a classifier system based on XCS-LP, but it is configured for just one, dummy, action. In this configuration $a_k^*$ is treated formally like a *payoff to be learned*, with $\hat{a}^*$ the predicted value. Thus S2 in effect acts as a function approximator approximating $a^*(x)$. If S2's error criterion $\epsilon_0$ is small (strict) then S2's approximation $\hat{a}^*$ will be similar to $a_k^*(x)$ which is a staircase-like function of $x$. However, if $\epsilon_0$ is not small, S2's approximation will be less discontinuous, becoming smoother with larger $\epsilon_0$. Consequently, as $x$ changes continuously, $\hat{a}^*$ will also change continuously so that S2 will approximate $a^*(x)$, the optimal continuous policy for *Env*.

The paper will demonstrate the behavior of the S1-S2 system using the so-called "frog" problem [1]. In the frog problem, $x$ is the sensory stimulus from a "fly" at a distance $d$, $\{a_k\}$ are discrete jump lengths that the frog can take, and the payoff $R$ is the sensory signal following the jump. While the $a_k^*$ are clearly discrete since the $a_k$ are discrete, $a^*(x)$ for this problem is continuous and smooth (in fact linear). The paper will show how S2's prediction $\hat{a}^*(x)$ ranges from discontinuous (staircase-like) to continuous and linear, depending on the setting of S2's $\epsilon_0$.

It is hoped that the S1-S2 architecture will contribute at least conceptually and possibly practically, to solution of the difficult problem of learning optimal continuous policies.

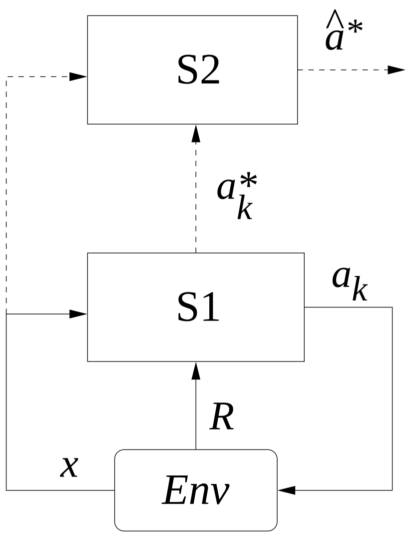


**Fig. 1.** Classifier system architecture for learning optimal continuous policy.

## References

1. Stewart W. Wilson. Classifier systems for continuous payoff environments. [GECCO-2004].