Optimal Continuous Policies
A Classifier System Approach

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

Typically, how does an LCS (RL) learn?

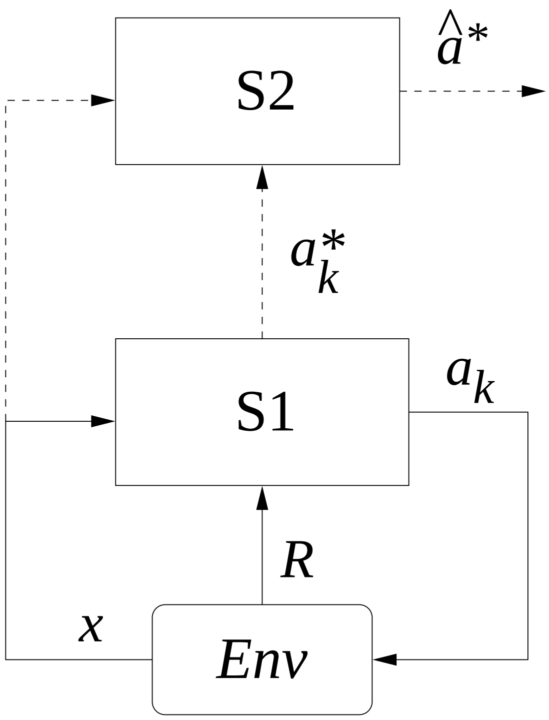
- Learns the values of actions. $P(x, a)$, $[Q(x, a)]$
- Picks the best action—call it $a^*(x)$.
- Does not learn $a^*(x)$ directly.

Why not learn the best action directly?

- It's hard.
- Given payoff P for a and x, what do you do?
- Not clear even if relative P is known.
- That's why RL first learns all $P(x, a_k)$.

What are you suggesting?

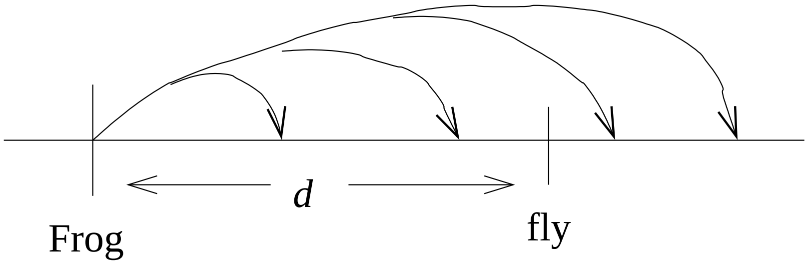
A simple approach, possibly not new.



- Essential idea: Upper system **observes** lower system, learns best action.

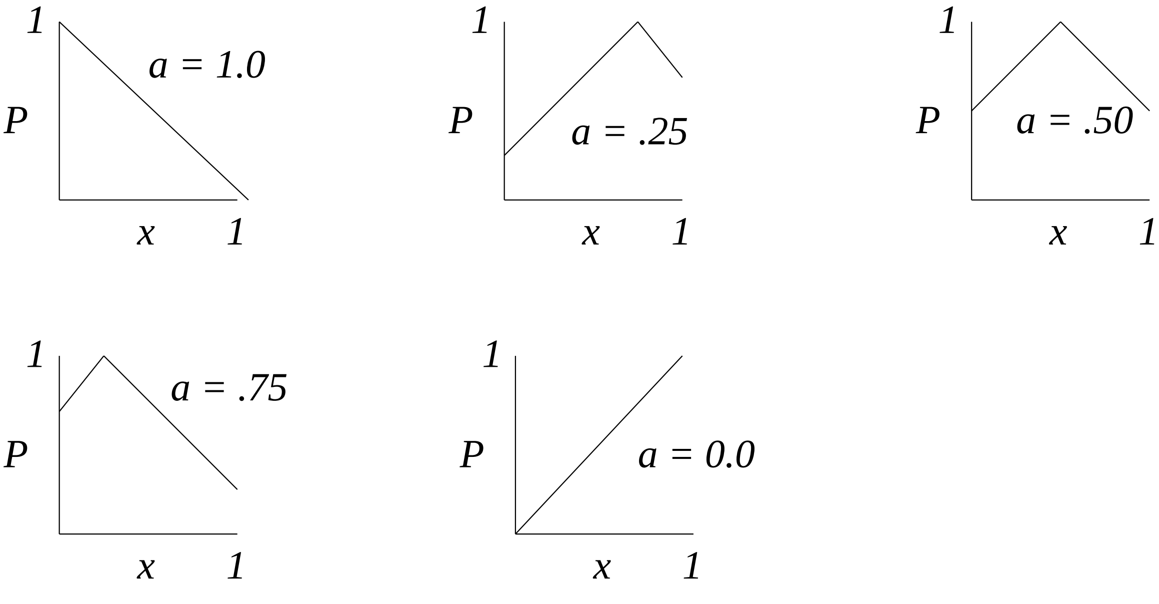
- Also permits actions to be continuous-valued.

“Frog” problem



$$x = 1 - d \qquad 0.0 < d < 1.0$$

$$P = 1 - d'$$



What do lower system classifiers look like?

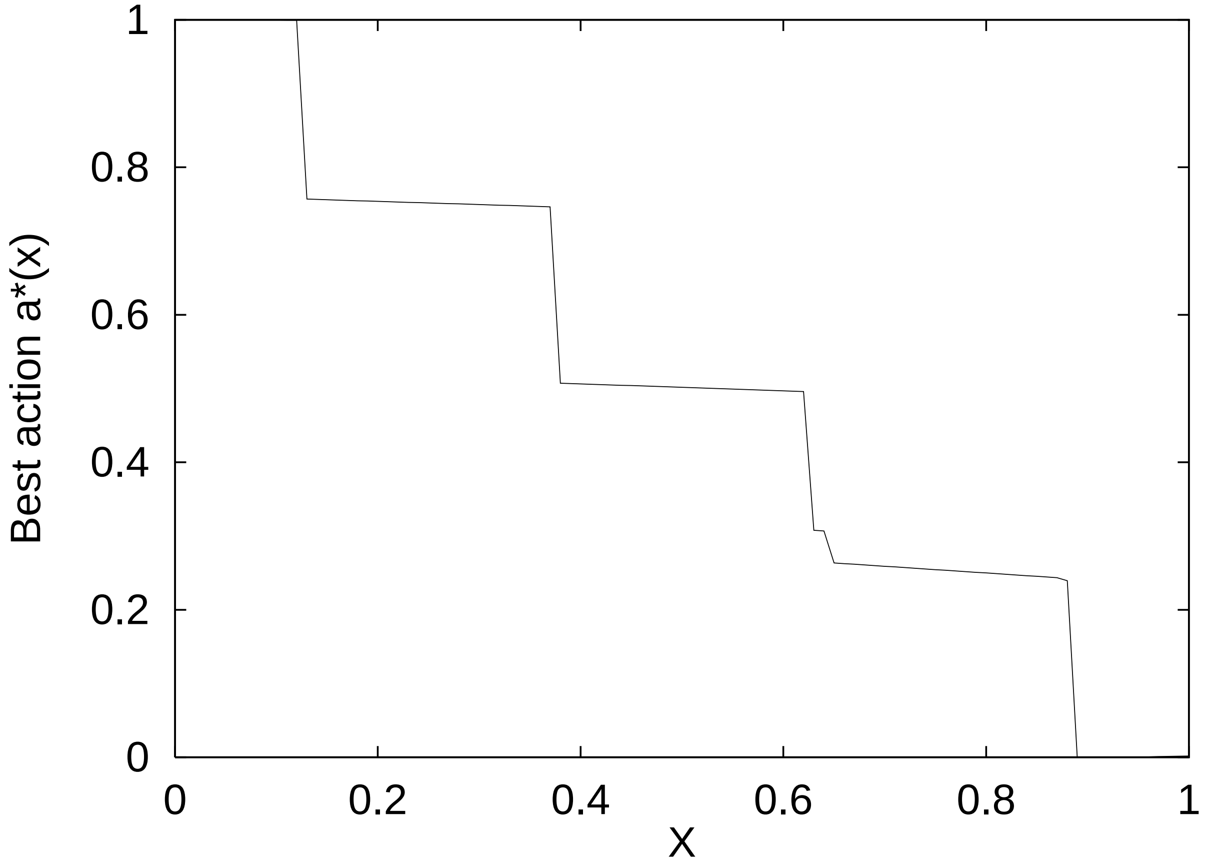
Form: $t(x) : a \Rightarrow w \cdot x'$

Example: $(.00, .75) : .25 \Rightarrow 0.25 + 1.0x$

- Condition evolves using GA exactly as in XCS.
- Prediction weight vector is adapted using a modified delta rule.
- Result: Condition and weight vector *co-evolve* to minimize the classifier's prediction error, and maximize the subspace it matches.
- In a problem with piecewise-linear payoff (such as the Frog problem), the classifiers of XCS-LP should evolve to correspond exactly to the segments of the payoff functions.
- In a general non-linear problem, the classifiers should evolve to form a piecewise-linear *approximation* to the payoff functions.

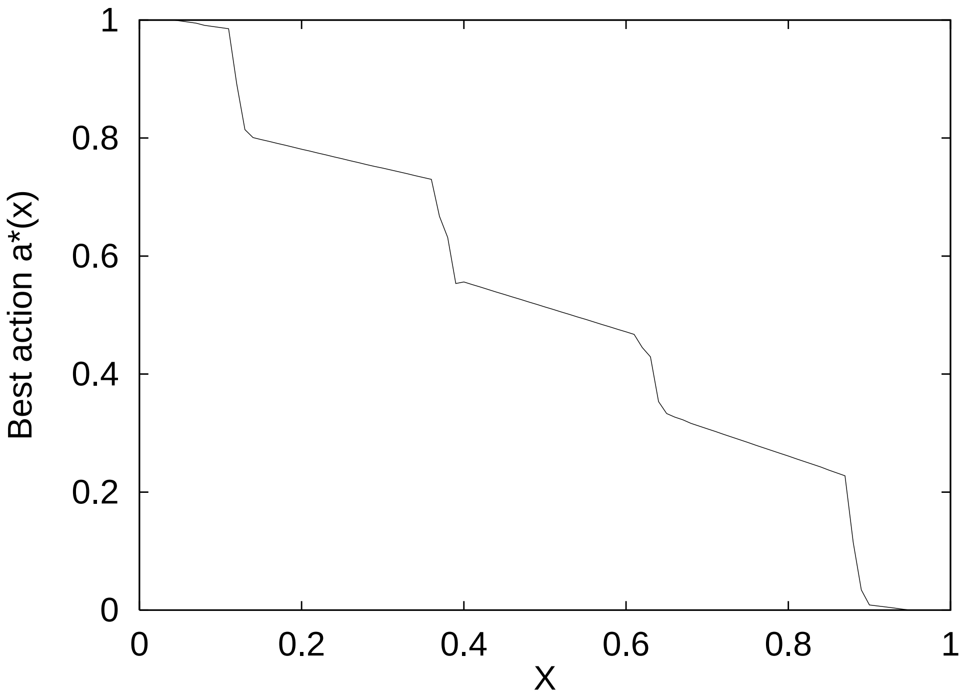
What is the upper system?

- A classifier system but with one, null action.
- Typical classifier: $t(x): \;\; \Rightarrow w \cdot x'$
- Upper system approximates the best-action values, $a^*(x)$, of the lower system. $a^*(x)$ is the upper system's "payoff".
- Graph of result:



How do you get continuous actions?

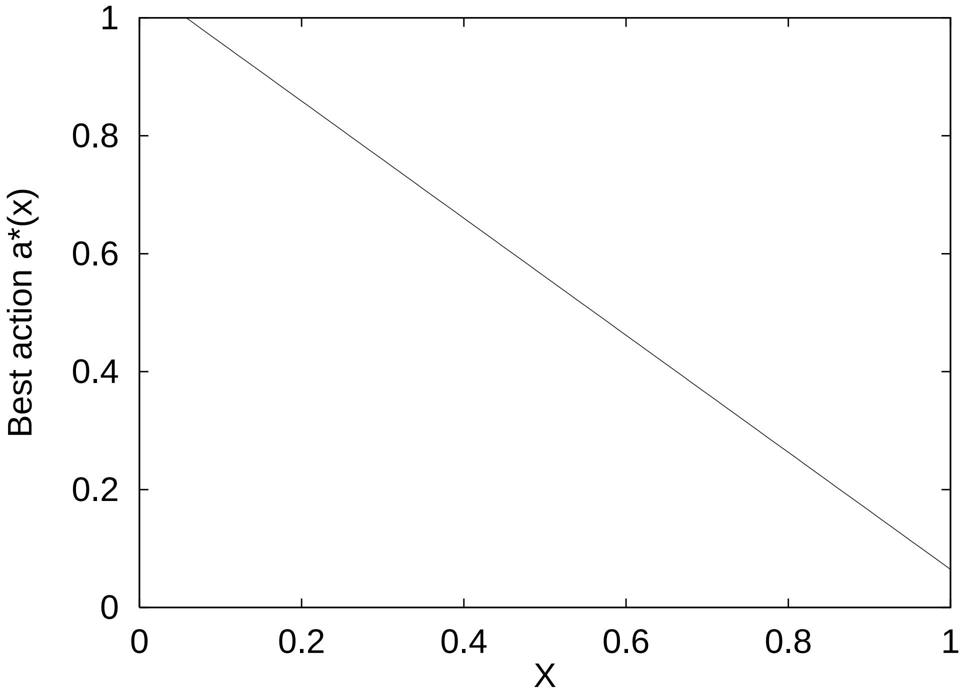
- Relax error threshold ϵ_0 of upper system.
- Classifiers with more general conditions win out over more specific conditions so that sharp corners begin to round off.



Can you get a straight line?

- Large enough ϵ_0 allows a single classifier to take over the population.

$$(0.0, 1.0): \quad \Rightarrow 1.06 - 0.99x$$



"If fly stimulus is x, jump $1 - x$".

Where are we?

- Possible route to continuous policies/actions.
- Relation between ϵ_0 and "bandwidth" of $a^*(x)$.
- Lower system samples the action space using fixed, discrete actions. Better schemes may be possible.
- Though awkward, upper/lower system helps conceptualize the continuous action problem.
- Continuous actions essential for realistic robotics and organism modeling.