Classifier Systems for Continuous Payoff Environments

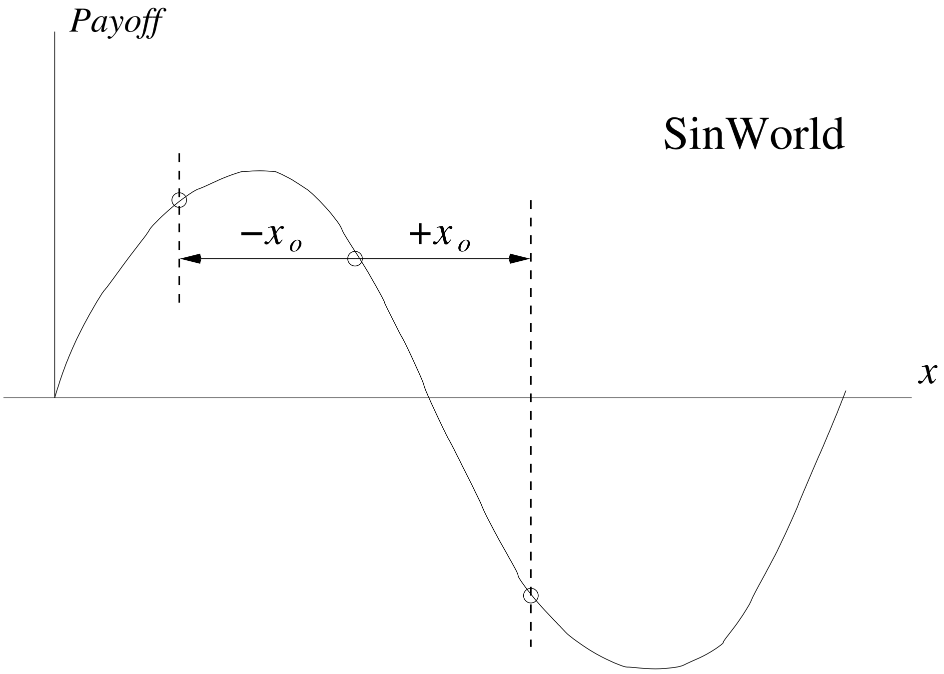
Stewart W. Wilson
Prediction Dynamics, Concord, MA 01742
Department of General Engineering
University of Illinois at Urbana-Champaign IL 61801
wilson@prediction-dynamics.com

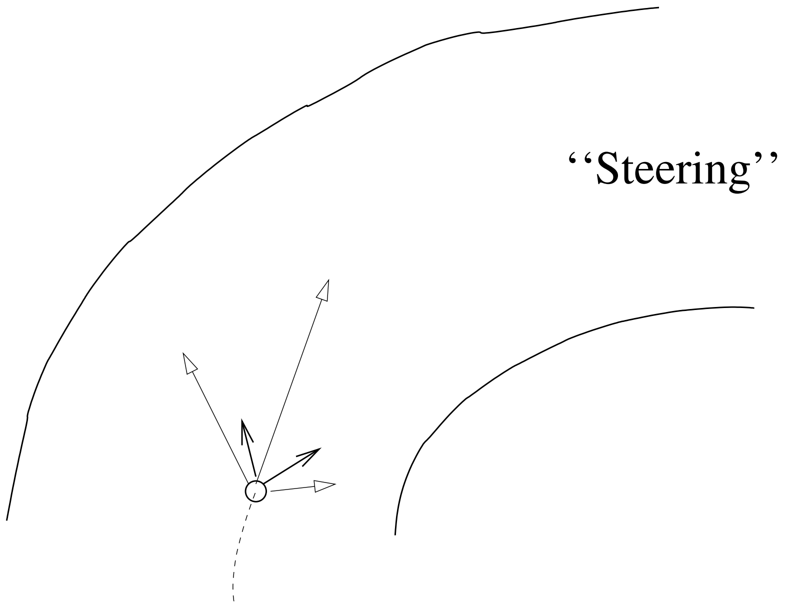
What are traditional LCS payoff environments like?

Boolean functions, Data inference: $dP \neq k \cdot dx$

$P(x, a)$ is step-wise discontinuous.

Continuous payoff environments?





What Kind of Classifier?

Usual syntax

$1\#00\#\#10 : 1 \Rightarrow 934$

or

$(.10, .92)(.34, .57) : 0 \Rightarrow 560$

i.e.,

$t(x) : a \Rightarrow p$

But consider

$t(x) : a \Rightarrow p(x)$,
where the prediction is a *function of* x.

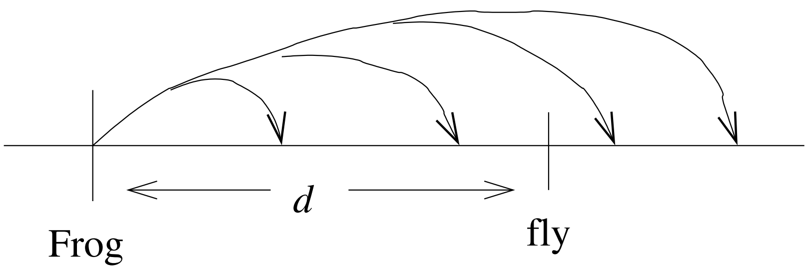
In particular,

$t(x) : a \Rightarrow w \cdot x'$,
with w a weight vector.

This classifier syntax is used in XCS-LP (linear prediction).

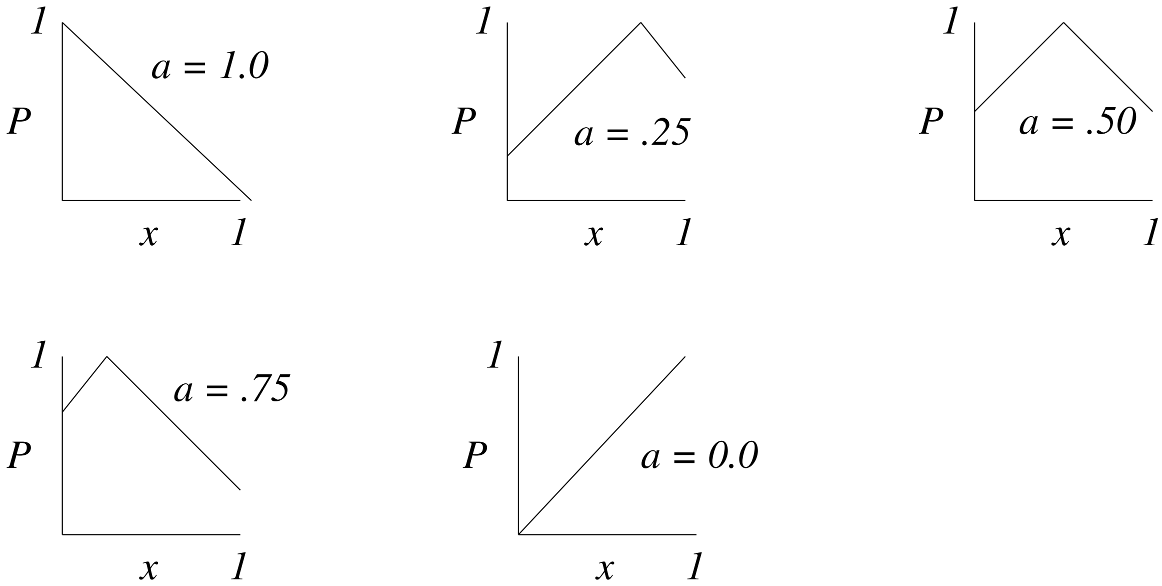
The purpose is to provide a more compact and transparent representation in problems with continuous payoff functions.

"Frog" problem



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

$$P = 1 - d'$$



What should frog classifiers look like?

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.

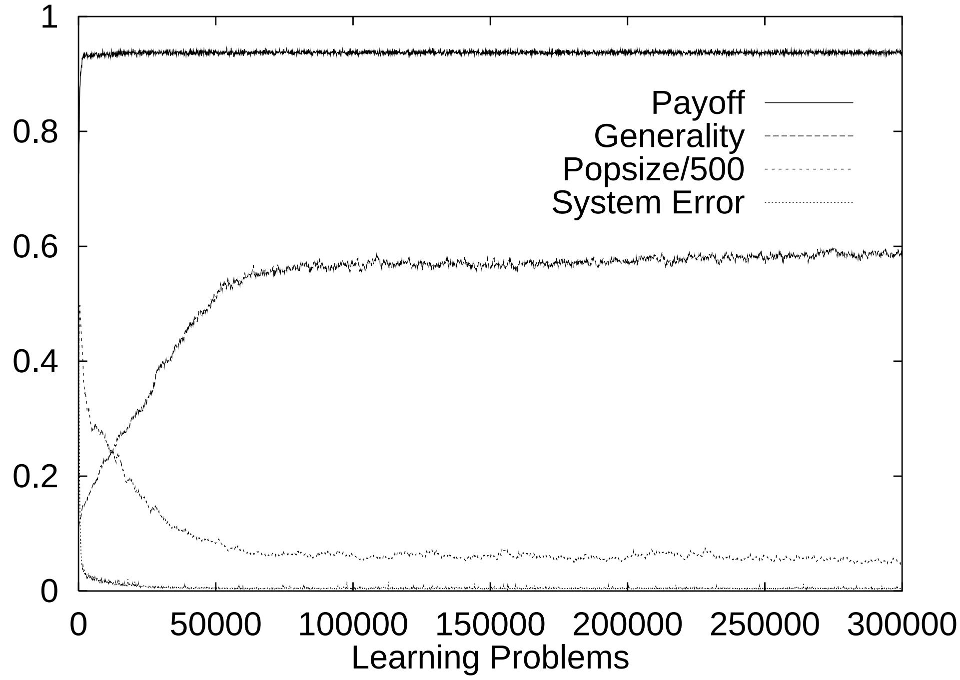
What do the frog classifiers actually look like?

```
     CONDITION     ACT     WTVECTOR      ERR    FITN  NUM EXPER

 0. |OOOOOOOOOO| 1.00   1.00 -1.00   0.000  .997  76  28224
 1. |OOOOOOOOOo|  .00   0.00  1.00   0.000  .970  67  8274
 2. |....oOOOOO|  .50   1.49 -0.98   0.003  .916  56  7463
 3. |OOOOOo....|  .50   0.50  0.98   0.003  .769  48  3057
 4. |........oO|  .25   1.46 -0.69   0.010  .902  40  2584
 5. |OOOOOOOo..|  .25   0.25  0.99   0.007  .567  38  21004
 6. |OOo.......|  .75   0.76  0.95   0.003  .942  38  1450
 7. |........oo|  .25   1.47 -0.68   0.013  .260  28  2999
 8. |..oOOOOOOO|  .75   1.23 -0.98   0.005  .555  26  5231
 9. |OOOOOOOo..|  .25   0.25  0.99   0.007  .364  23  801
10. |.......oo.|  .25   1.50 -0.68   0.007  .478  21  880
11. |..oOOOOOOO|  .75   1.23 -0.97   0.013  .044  20  1088
12. |..oOOOOOOO|  .75   1.25 -1.00   0.000  .335  11  3776
13. |.........o| 1.00  -0.99  1.01   0.013  .000   1  6
14. |.........o|  .25   1.36 -0.65   0.198  .000   1  12
15. |.........o|  .50   1.23 -0.78   0.180  .000   1  10
16. |........oO|  .25   1.47 -0.69   0.014  .001   1  39
17. |OOOOOOOOo.|  .25   0.25  0.99   0.007  .017   1  133
18. |.........o|  .75  -0.33 -0.34   1.658  .001   1  4
19. |.........o|  .00  -1.00 -1.00   0.005  .002   1  0
20. |OOOOOo....|  .50   0.50  0.98   0.013  .008   1  22
```

- Entire population!

What about performance?



- High performance reached quickly.

- Performance is optimal, given discreteness of actions.

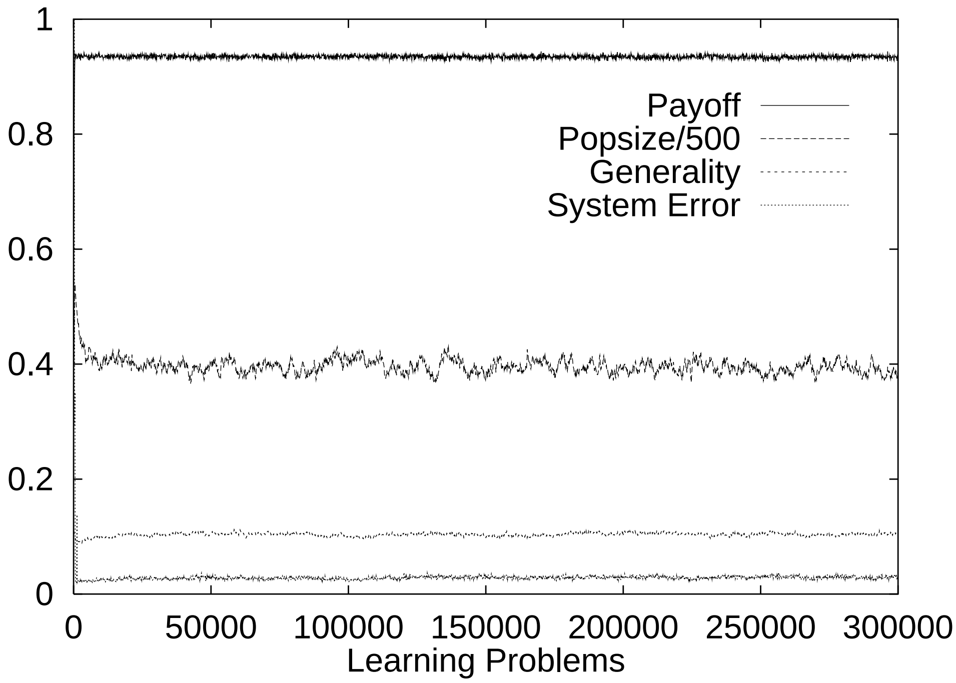
- Condensed population reached by 50,000 problems.

How about standard XCS on the frog?

```
    CONDITION    ACT    PRED   ERR   FITN   NUM EXPER

 0. |o.........| 1.00   .948  .011  .601   11  167
 1. |oo........|  .50   .608  .020  .252    8  232
 2. |..o.......|  .00   .238  .008  .551    7  46
 3. |........oO|  .75   .332  .022  .148    7  83
 4. |......o...|  .75   .587  .008  .461    7  23
 5. |........oo|  .25   .845  .026  .208    7  37
 6. |....oo....|  .50   .982  .013  .286    7  39
 7. |......oOo.|  .25   .940  .023  .224    7  46
 8. |oOOOo.....|  .75   .911  .036  .424    7  82
 9. |..oo......|  .00   .319  .018  .238    7  15
10. |..oOo.....|  .25   .552  .035  .385    7  11
```

- Top 11 of 181 classifiers.



What's the point?

- For continuous payoff functions, the linear predictions of XCS-LP allow smaller populations and more transparent "knowledge" than do the simple scalar predictions of XCS.

- Advances the ability of classifier systems to evolve compact, and at the same time readable, representations. I.e., to **generalize**.

- Point of view: regard prediction as a function and then approximate it.

- Look at other approximation bases and parameter adjustment techniques.

- XCS-LP should work with any nonlinear, piecewise-continuous payoff function. Works with SinWorld.