

Learning Classifier Systems

A Brief Overview of the First 30 Years

Overview

- Holland's idea
- Some early successes but
- Wilson's XCS
- Encouraging results
- A renaissance
- Supervised learning
- Regression
- Unsupervised learning
- Next

Context

- Back in the 60' and 70's the main thrust of AI was in the development of expert systems.
- Here a given domain expert's knowledge is translated into if-then rules and executed on a computer.
- However, several drawbacks soon became apparent:
 - the rules might be incorrect
 - the rules might not be self-consistent
 - the rules might not capture all of the domain space
 - experts don't always agree
 -

A Better Way

- Thus the early (?) expert systems were complicated to build and typically very brittle.
- One of the dominant features of biological intelligence is its adaptability.
- That is, organisms change their behaviour in response to changing circumstances.
- Early pioneers like Turing had highlighted this fact.

From GAs

- In his famous 1975 book, Holland reasserted how intelligent behaviour in artificial systems might be achieved through simulated evolution.
- After describing the canonical binary Genetic Algorithm, he presented the Broadcast Language.
- Essentially, he proposed using genetic operators to design strings of symbols which detected an environment and produced outputs (and possibly new strings too).
- The GA would design the strings to learn what to detect and how to respond to increase fitness.

... to LCS

- In 1978, in collaboration with Judith Reitman, Holland presented “Cognitive System 1”.
- Essentially, a cut down version of the Broadcast Language with a credit allocation strategy to assign fitness to individual strings/rules.
- CS-1 was shown able to navigate a one-dimensional maze optimally.
- A revised version was presented in 1980 which became the “standard” form for the following 15 years.

In a nutshell

- Rule-based (production system).
- Use **reinforcement learning** techniques to assign utility to rules.
- Use **evolutionary computing** techniques to discover new rules.
- Can incorporate other heuristics and/or prior knowledge to improve their performance.

Traditional Rule Syntax

- Each rule (classifier) is a state-action pair.
- IF <condition> THEN <action>
- Normally uses a trinary representation.
- E.g., 01#11 : 010
- Where # is a “wildcard” facilitating *generalization* - the rule above considers inputs 01111 and 01011.

Rule Fitness (strength)

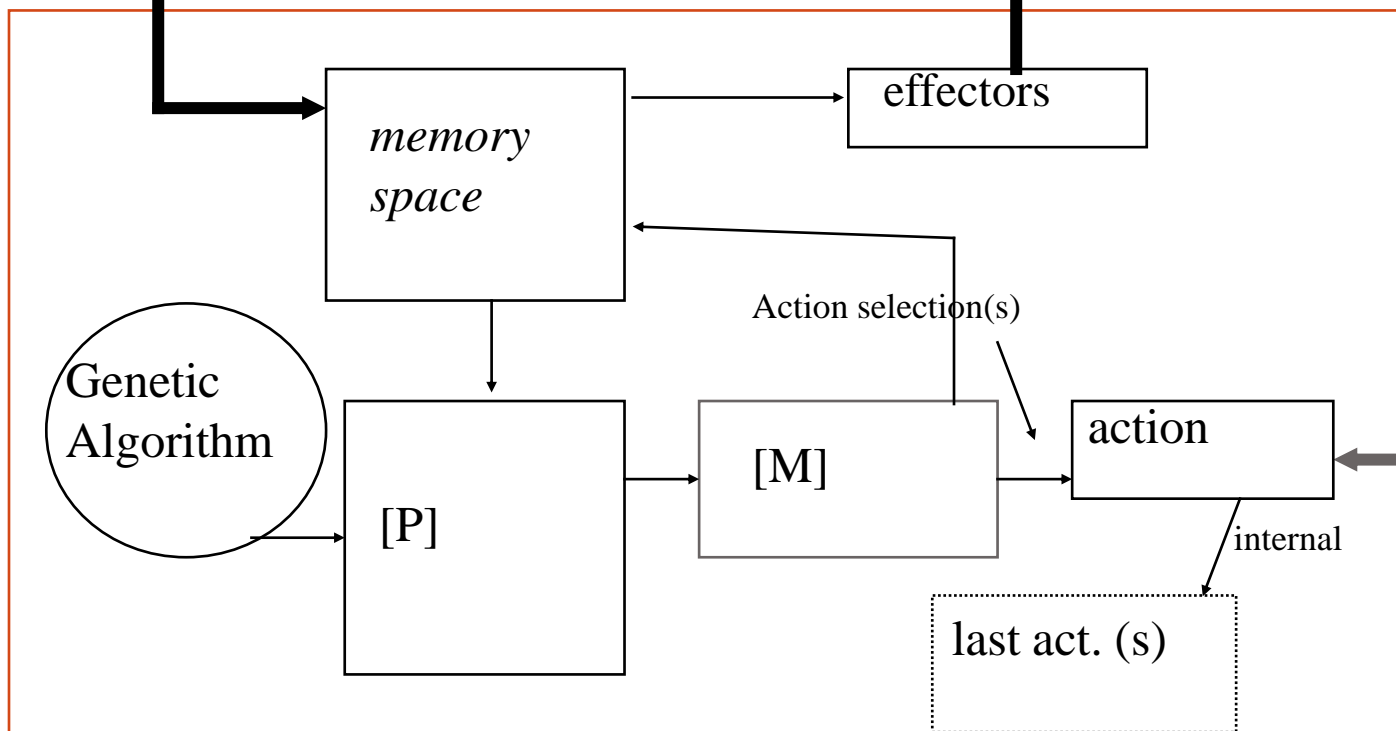
- Associated with each rule is a *fitness* parameter.
- The value of this parameter indicates the *external payoff* from using the rule.
- Adjusted during LCS interaction with the environment.
- Used in the *action selection* auction.
- LCS tries to maximize the reward it receives.

Environment

Input

Output

Payoff



Bidding

- The *bucket brigade* is a reinforcement learning algorithm.
- At each time step t a matched classifier C uses its fitness F and specificity S to bid B to have its action used as the LCS output:

$$B_C^t = k \cdot S_C \cdot F_C^t$$

- k is a constant $\ll 1$.
- For previous step rules: $F_C^{t+1} = F_C^t + \Sigma B^t / x$

Search

- Periodically, a steady-state GA is fired to produce new rules and delete less-fit ones.
- Hence the search for new rules is guided by current rules which receive higher payoff from the environment.
- Over time the rule-base comes to converge upon a *cooperative* set of effective rules for the learning task.
- Fitness sharing is the principle mechanism for niche maintenance.

Some Early Success

- David Goldberg (1983) was first to apply Holland's LCS to a real-world problem – gas pipeline control. LCS had to learn to balance flow in the face of varying demand and leaks.
- Stewart Wilson (1985) used a version of CS-1 to control a video camera such that objects were centred in its view.
- Marimon et al. (1990) modelled agents in artificial markets.
- Frey and Slate (1991) showed competitive performance on a letter recognition task.

... it didn't last

- Half of the papers at the 1985 International Conference on GAs were about LCS.
- Ten years later, there were just two at the same bi-annual event.
- Wilson and Goldberg wrote an overview in 1989 which highlighted how the formation of rule chains had proven most problematic.
- They proposed some ways forward but only a few people continued to work with LCS.

Stewart Wilson's XCS

- One of the people to continue with LCS was Wilson.
- He had concentrated on simplifying Holland's algorithm.
- In 1995 he presented the eXtended Classifier System.
- XCS makes two important changes:
 - Rule fitness is based on the accuracy of its predicted payoff
 - The bucket brigade is replaced by a form of Q-learning

Why Accuracy?

Input	Output	Reward
0	0	1000
0	1	2000
1	0	1000
1	1	800
#	0	1000
#	1	1400

Input: 1

Opt. Action: 0

Σ **Action: 0 = 2000**

1 = 2200



An *overgeneral* with high average payoff
but also low accuracy (2000 \longleftrightarrow 800).

Environment

Payoff

detectors

effectors

[P]

GA,
covering

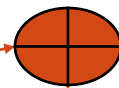
Action selection

[A]

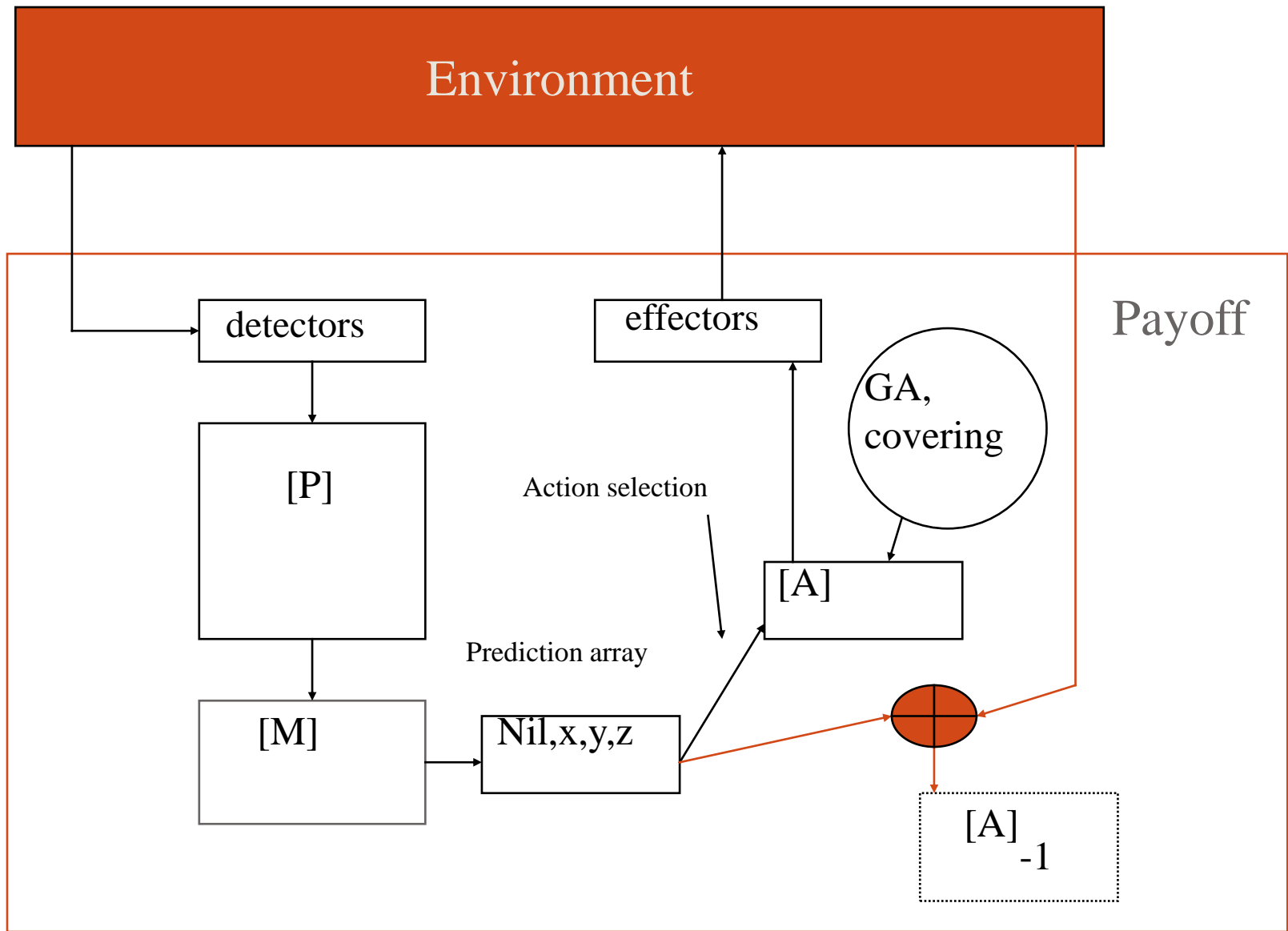
Prediction array

[M]

Nil,x,y,z



[A]
-1



Rule Parameters

- Predicted payoff, p .
- Prediction error, ε .
- Fitness, F .
- Numerosity, n .
- Estimate of niche size, a .
- Time stamp of last GA activation.

Parameter Updating

- Updating is essentially a five-step process.
- It begins by updating each rule's *error* (ε) using the Widrow-Hoff delta rule:

$$\varepsilon \leftarrow \varepsilon + \beta (|R - p| - \varepsilon)$$

- Then the *prediction* value is adjusted:

$$p \leftarrow p + \beta (R + \gamma \max(P_{[A]_{+1}}) - p)$$

From error to fitness

- Then the *accuracy* (k) is computed:

$$k = 0.1 (\varepsilon / \varepsilon_0)^{-\nu}$$

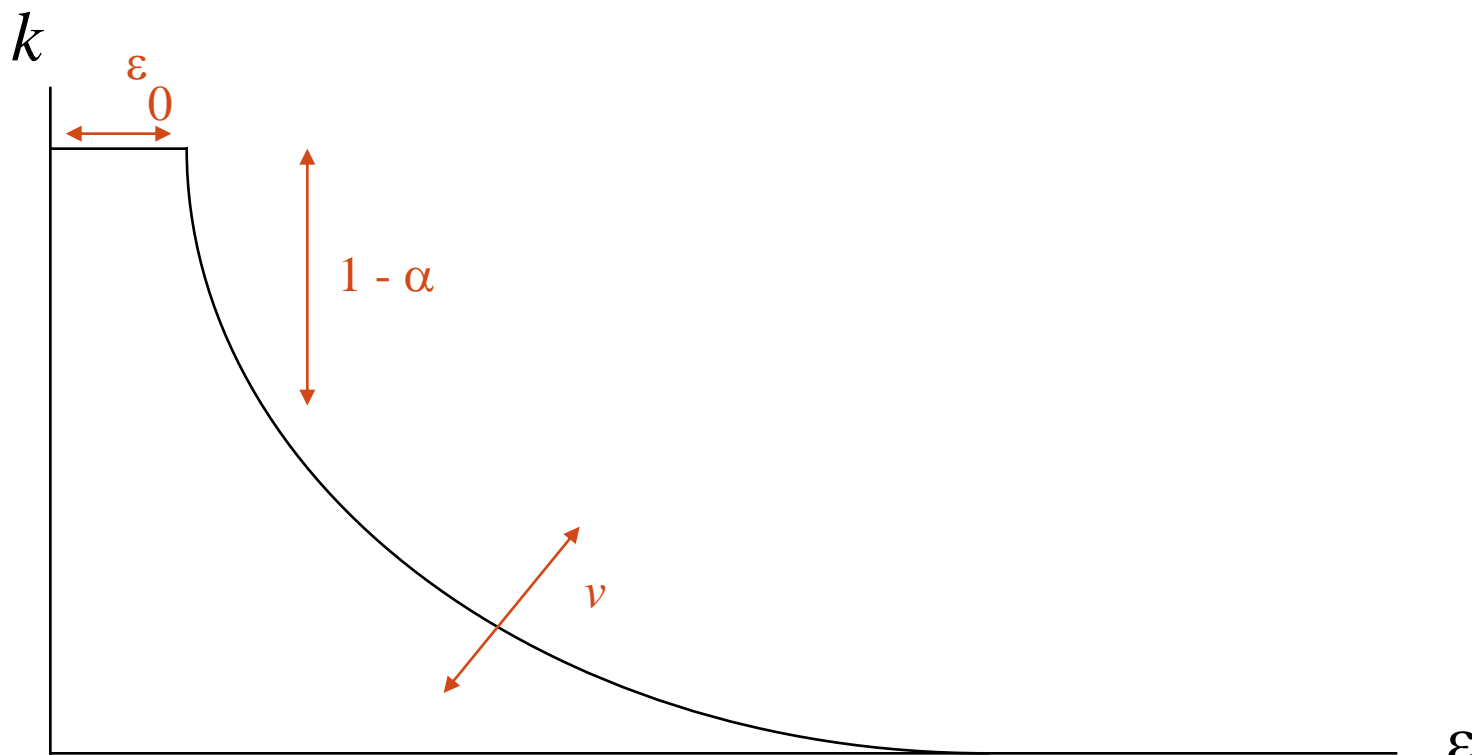
- For $\varepsilon > \varepsilon_0$, else $k = 1.0$. $\nu = 5$, $\varepsilon_0 = 0.01$.
- The function falls off exponentially for $\varepsilon > \varepsilon_0$
- Then *relative accuracy* (k') is calculated:

$$k' = k / \Sigma k$$

- Finally, *fitness* is updated:

$$F \leftarrow F + \beta (k' - F)$$

Visually



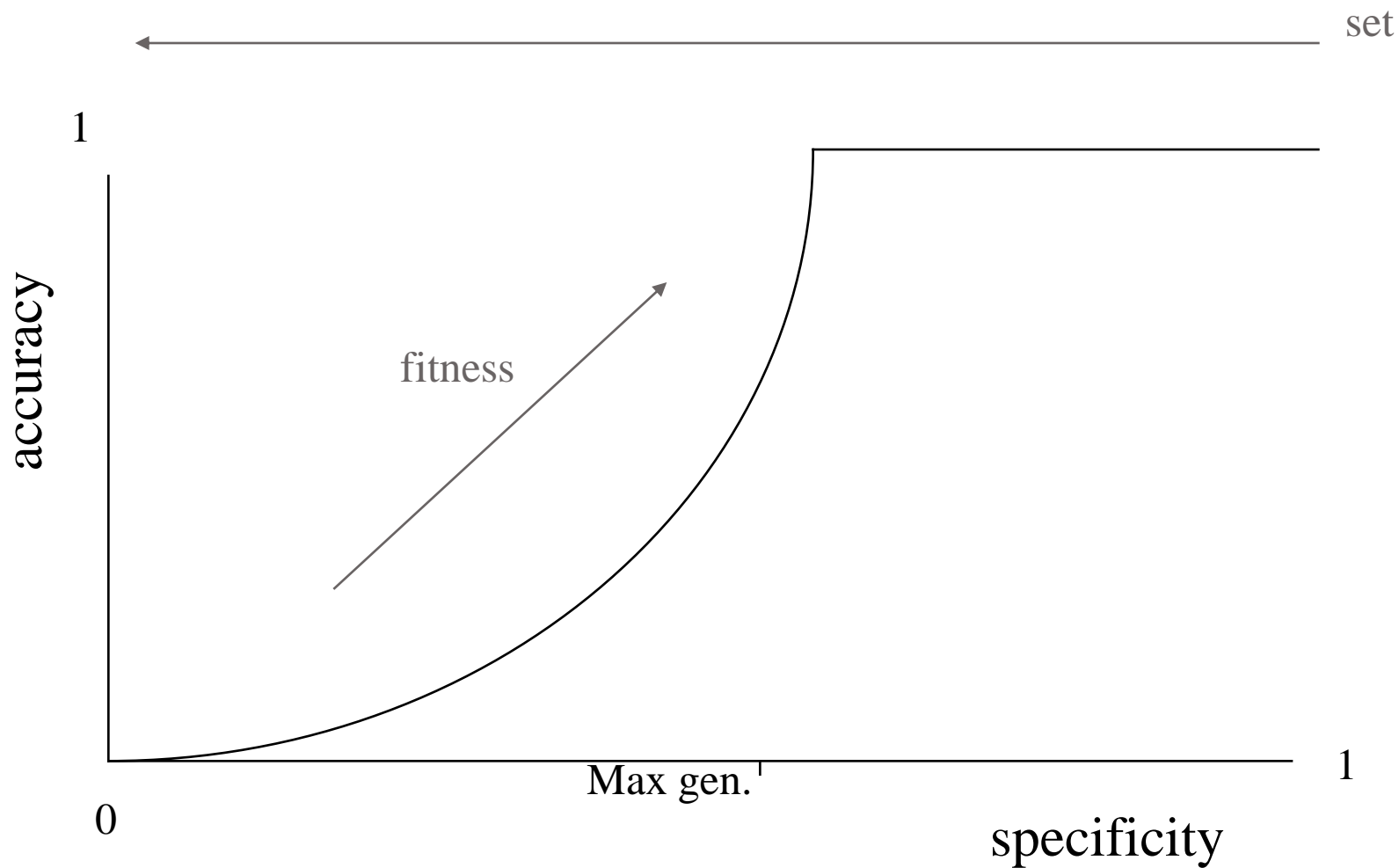
Search

- Rule fitness is based on the accuracy of payoff predictions.
- Rule deletion is based on niche size.
- Thus XCS attempts to build a complete and accurate payoff map for a given problem (high and low).
- And the rulebase resource is balanced across all areas.
- Essentially, it uses rules to represent a compacted form of the equivalent Q-table.

Early Results

- Wilson showed optimal performance on two well-known benchmark problems.
- Perhaps equally significantly, the solutions learned were maximally general (for the given encoding).
- That is, the payoff map was held in the fewest possible rules.
- Subsequently, there is a growing body of theory about how XCS learns.
- There are two basic pressures within XCS.

Maximal Generality



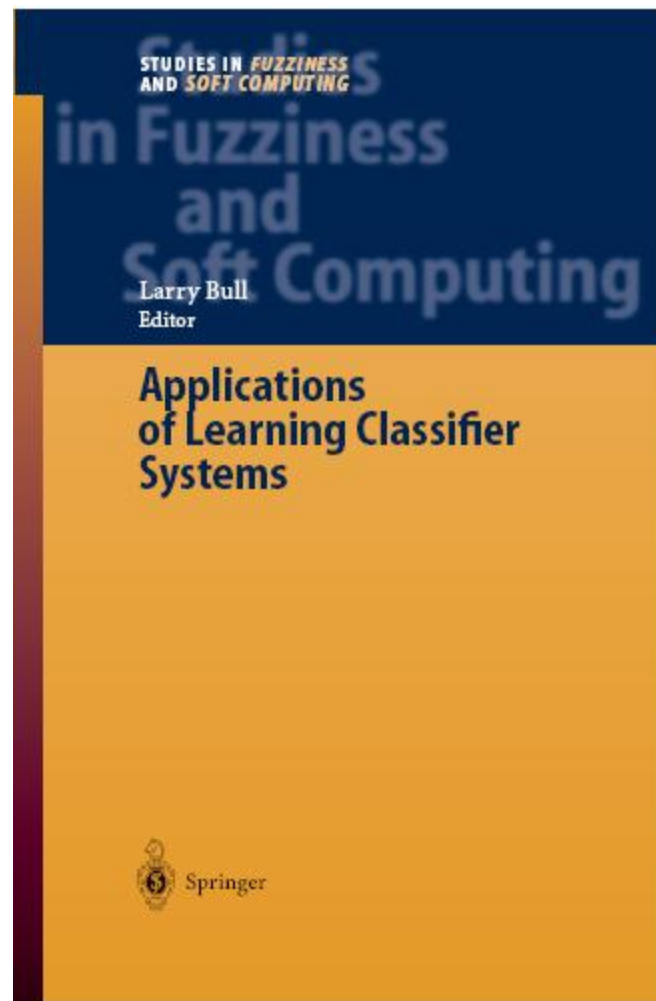
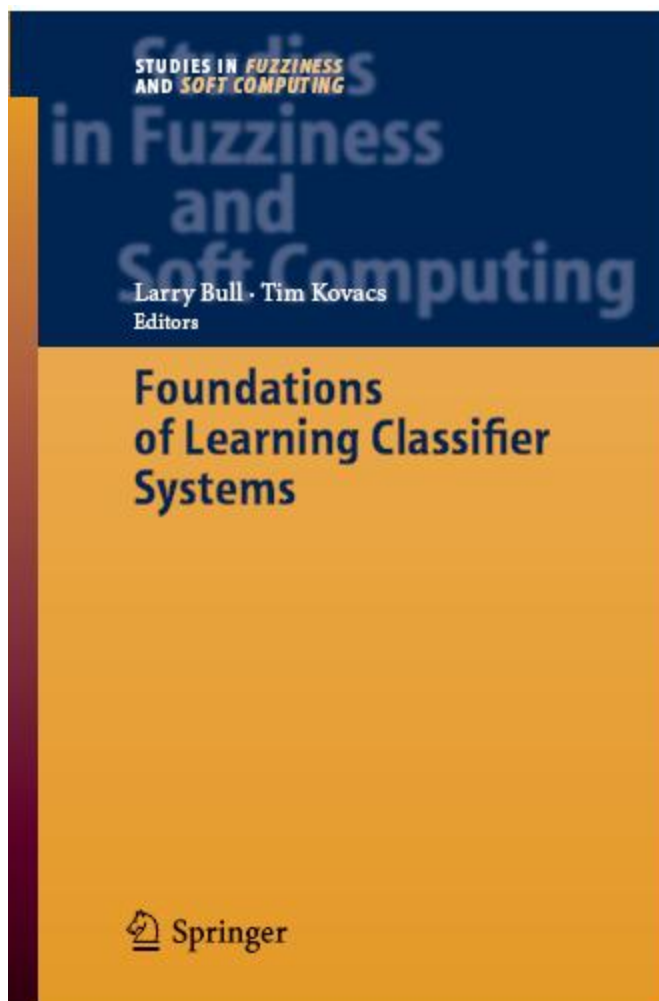
A renaissance

- Very few people took any notice of XCS for some years but interest did begin to grow.
- At the first Gecco conference in 1999, a workshop was held on LCS – the IWLCS.
- Since then it has been held annually alongside Gecco and an increasing number of people are using XCS and its derivatives.
- A significant amount of subsequent work has explored incorporating modern mechanisms from evolutionary computing and reinforcement learning into XCS, along with some fundamental LCS research.

Some Key Works

- Anticipatory Learning (e.g., Stolzmann, GP1998)
- GP (e.g., Lanzi & Colombetti, Gecco 1999)
- Self-Adaptation (e.g., Bull et al., SAB 2000)
- Memory (e.g., Lanzi & Wilson, ECJ 2000)
- Continuous Time and Space (e.g., Hurst et al., PPSN 2002)
- ANN (e.g., Bull, PPSN 2002)
- Local Search (e.g., Wyatt & Bull, MA 2004)
- Gradient Descent (e.g., Butz et al., Gecco 2005)
- EDA (e.g., Butz & Pelikan, Gecco 2006)
- Multiple Objectives (e.g., Studley & Bull, ALifeJ2006)
- Bayesian (e.g., Dam et al., IEEE J 2006)
- Fuzzy Logic (e.g., Casillas et al., IEEE JFUZ2007)
- Tile-Coding (e.g., Lanzi et al., Gecco 2007)

Some Theory and Practice



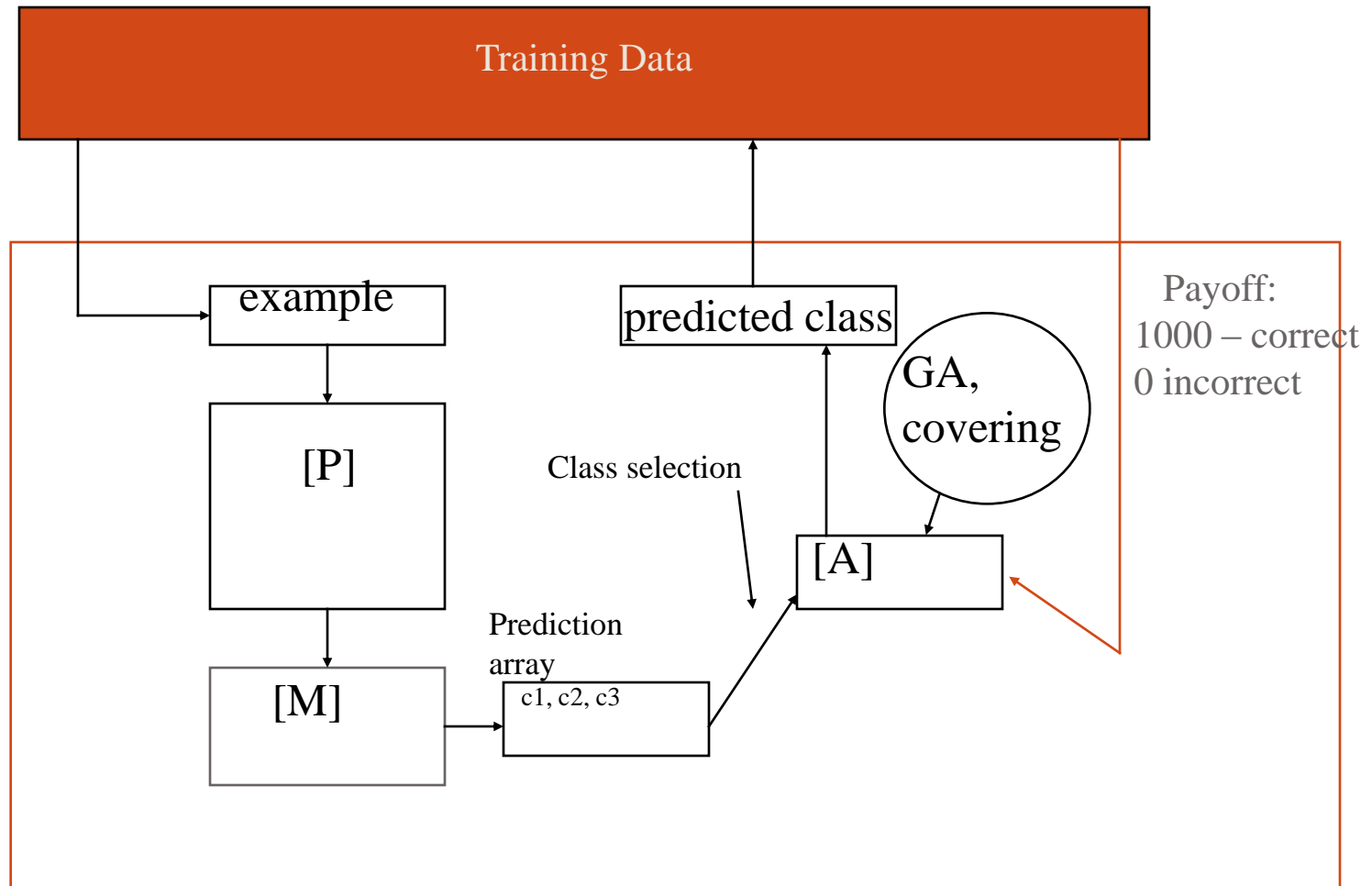
Data Mining

- Since XCS is rule-based and it learns maximally general descriptions of a problem space, its potential for data mining was recognised almost immediately.
- Wilson introduced interval encodings and reported competitive classification accuracy on the Wisconsin Breast Cancer data set (IWLCs 2000).
- Alwyn Barry used XCS for its first commercial application – directed marketing - soon afterwards.

Interval Encodings

- Wilson has introduced two versions of the interval encoding to enable LCS to handle integers and reals.
- Here rule condition elements are ranges defined by either a centre with a distribution, e.g., (8.05, +/-1.03), or by an upper and lower bound, e.g., (7.02, 9.08).
- A given condition element is said to match an attribute x when $l \leq x \leq u$
- Crossover occurs between and within ranges.
- Mutation adds/subtracts a value generated from a uniform random distribution of determinable size $\{0, max\}$.

XCS Data Mining Schematic



Primary Breast Cancer Case Study

- Primary breast cancer diagnosis is a major challenge to oncologists who treat breast cancer since it is the first stage from where the cancer develops.
- The Frenchay Breast Cancer (FBC) dataset is a real-domain dataset which has a description of pathological data for women with primary breast cancer.
- The accuracy of XCS was found to be 80.1% +/- 5.9
- C4.5 was found to give an accuracy of 77.4% +/- 3.3
- The rules learned by both approaches were then presented to a doctor for scoring.
- XCS found 9 rules suggested as representing new knowledge. C4.5 found 4. (Kharbat et al., Gecco 2007)

Example Rules

IF Immuno pS2 score<=0.879

THEN the Grade= G1 (197\0)

IF age>=30.19
late2a1ity =B
S10chro0ous Tumour? =false
DCIS Necrosis =true
DCIS component =true
Size of DCIS + Invasive>=23.91
Ex2ision M0rgin 2o3e =C
Excision Margin mm>=1.517
Involved nodes total>=1.503
Extra-N0dal Invasi0n =true
Immuno Done? =true
Immuno ER H Score>=123.632
Immuno PR pos =true
Immuno PR score>=78.55
Sum>=4.9206218734013945
THEN the Grade= G3 (3\0)

Supervised Learning

- All of the early work using XCS for data mining maintained the reinforcement learning scheme.
- Wyatt et al. (e.g., ACDM 2002) fixed the prediction to that of the first training instance seen.
- Bernadó-Mansilla (e.g., ECJ 2003) has presented a form of XCS tailored to supervised learning for classification.
- The fitness of rules is simplified and only correct rules from within $[M]$ are evolved.

sUPervised Classifier System

- Rule accuracy k is now simply:

$$k = \frac{\text{number of correct classifications}}{\text{number of training matches}}$$

- After fitness updating, those rules whose action match the training example form the correct set [C].
- The GA is then run within [C].
- All other processing as in XCS.

UCS Performance

- The sUpervised Classifier System UCS has been shown able to learn faster than XCS.
- Also been shown to perform better with an increasing number of classes.
- It has been shown better able to handle unbalanced data sets.
- Now the LCS of choice for classification.

Regression

- Almost all LCS maintain a separate condition and action.
- Thus the action/output is not a direct function of the input.
- Wilson introduced a form of XCS designed for function approximation/regression tasks – XCSF (Gecco, 2001).
- Here a rule's action is computed by a linear combination of the input and a weight vector.

XCSF

- Action/output is computed for each rule:

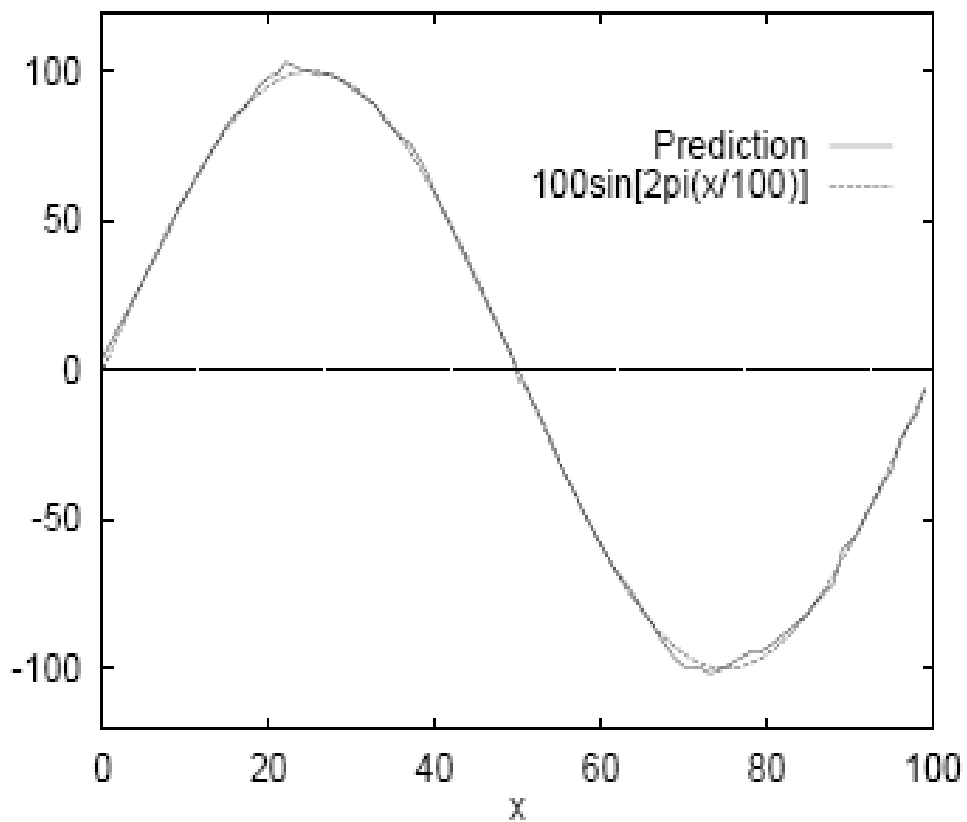
$$cl.w_0 \times x_0 + \sum_{i>0} cl.w_i \times s_t(i)$$

- Errors are updated based on difference between output and $f(x)$ using a modified delta rule:

$$\Delta w_i = \frac{\eta}{|\mathbf{x}_{t-1}|^2} (r - cl.p(s_{t-1})) x_{t-1}(i)$$

- All other processing as in XCS.

Example



CONDITION

```

0. |000000.....|
1. |00000o.....|
2. |00000.....|
3. |...0000o.....|
4. |.....o0000|
5. |.....o000000|
6. |.....000000000o....|

```

Unsupervised Learning

- Evolutionary algorithms have been used for clustering in a number of ways:
 - Use them to search for appropriate centres of clusters with established clustering algorithms such as the k -means algorithm, e.g., the GA-clustering algorithm. This typically requires the user to provide the number of clusters.
 - Treat the two aspects of the number and accuracy of clusters as multiple objectives.
- The generalization mechanisms of XCS can be used to evolve rules which describe maximally large clusters for a given accuracy threshold – XCSc (e.g., Tamme et al., Gecco 2007).

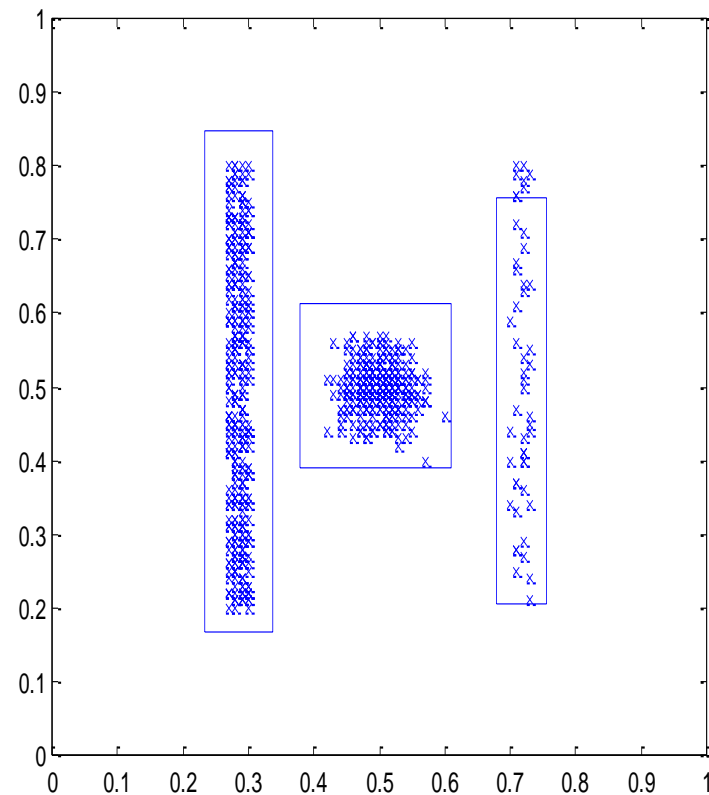
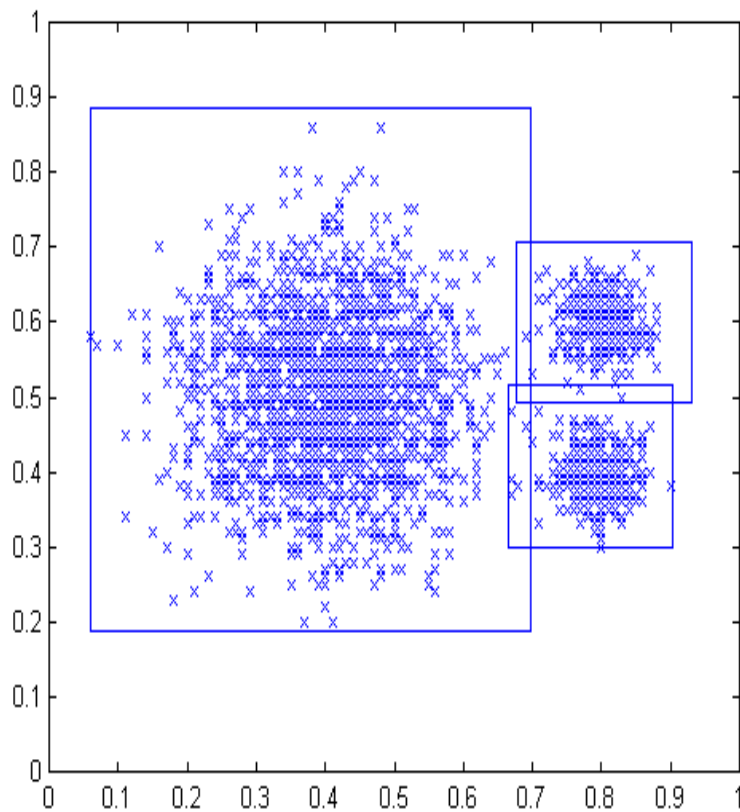
Fitness

- Use the centre-spread interval encoding $\{\{c_0, s_0\}, \dots\}$.
- Error ε is derived from the Euclidean distance with respect to the input x and c in the condition of each member of $[M]$:

$$\varepsilon_j \leftarrow \varepsilon_j + \beta \left(\left(\sum_{l=1}^d (x_l - c_{lj})^2 \right) \right)^{1/2} - \varepsilon_j$$

- All other processing as in XCS.
- Note number of clusters emerges with solution.

Examples



Next

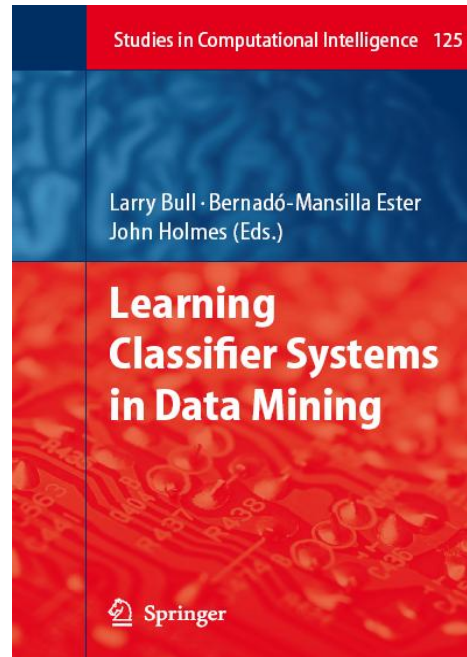
- Recent work has highlighted the parallels between LCS and ensemble machines.
- Ensembles of LCS have also been used.
- The online/incremental learning characteristic suggests they are particularly suited to very large and stream data sets.
- Improved knowledge discovery with richer rule representations and associated compaction algorithms.

Resources

- Martin Butz's book on XCS:



- New edited collection on LCS in data mining:



- The LCSWEB – <http://lcsweb.cs.bath.ac.uk/>