Slide 0

Part II — Evolutionary Learning

Jonathan Shapiro

Department of Computer Science, University of Manchester

June 18, 1999

Slide 1

A simple example — GABIL

Reference: De Jong

Overview: Learns a boolean concept by searching for a disjunctive set of propositional rules.

Representation of a rule: A substring for each variable in a rule, with a bit for every value which that variable can take plus a string for the concept value.

Example: if x takes values X_1, X_2, X_3 and y takes values Y_1, Y_2, then the string

$$110\ 01\ 1$$

is used to denote the rule

$$\text{IF } (x = X_1 \vee x = X_2) \wedge (y = Y) \text{ THEN } c = \text{TRUE},$$

where c is the concept.

Slide 2

Set of Rules: a long string of rule substrings concatenated together

Objective function: (number of examples correctly classified)^2

Crossover operator: allowed for variable-length rule sets.

- Two-point crossover used.
- Crossover points in second parent can be in any rule, but in analogous place in the rule as in the rule of first parent.

	rule 1			rule 2			rule 3		
	x	y	c	x	y	c	x	y	c
parent 1:	00|1	11	1	111	0|1	0	110	10	1

	rule 1			rule 2			rule 3		
	x	y	c	x	y	c	x	y	c
parent 2:	10|1	1|	1	0	001	01	0		

Slide 3

	rule 1			rule 2			rule 3		
	x	y	c	x	y	c	x	y	c
offspring 1:	001	11	0	110	10	1			
offspring 2:	101	11	1	111	01	1	001	01	0

Slide 4

Note that rules with more 1 are more general, e.g. all 1's matches any value.

Mutation: Biased towards introducing 1's.

- Operator which changes 0 to 1
- Operator which changes all bits associated with a variable within a rule to 1 (rendering that variable irrelevant in that rule).
- Ordinary unbiased mutation

Results: Compared with AQ14, ID5R, C4.5; found to give comparable results with unbiased mutation. Improved when biased mutation operators used.

Slide 5

Is the population the "answer" or is the best member of the population the "answer"?

Holland's original idea was to evolve rule-sets, where the rule-set would be the population.

- The "Michigan school"

DeJong evolved populations of rule-sets (like most work nowadays)

- The "Pittsburgh school"

Next we see some examples in which the entire population is used as the answer.

Slide 6

Time-series prediction

Reference: Packard, 1990, Meyer and Packard, 1992.

Task: Given a value of some quantity x now, and some number of time-steps ago, predict the value of x a time T in the future. I.e. given the *lag vector*

$$(x_t, x_{t-\tau}, \ldots, x_{t-n\tau})$$

predict x_{t+T}.

Modified Task: Recognise *if* prediction is possible; only then do prediction. Look for the "windows of predictability".

Purpose: Find a population of rules which are predictive. For situations where no rule in the population is satisfied, prediction may not be possible.

Slide 7

Rule Set: Conjunctions of ranges (variable number of terms), e.g.

$$\{(3.2 \leq x_6 \leq 5.5) \wedge (0.2 \leq x_2 \leq 0.3) \wedge (3.4 \leq x_{10} \leq 9.8)\}$$

Objective Function:

$$-\log\left(\frac{\sigma}{\sigma_0}\right) - \frac{\alpha}{N_c}$$

- σ is the standard deviation of the value T steps in the future for all data points which satisfy this rule;
- σ_0 the standard deviation T steps in the future for all data
- N_c is the number of points which satisfy the rule;
- α is a constant.

Slide 8

Selection: Rank the population by fitness; throw out and replace the low fitness ones.

Crossover: Uniform crossover, taking one range condition as a bit.

Mutation: Four types used:

- Add a new condition
- Delete a condition
- Broaden or shrink a range
- Shift a range up or down.

Results: Tested on the Mackey-Glass equation, a chaotic, non-local dynamical system. Found rules with small variances.

Slide 9

Learning Classifier Systems

Reference: Holland, 1986

Holland proposed a learning system based on his hypotheses about the efficiency of genetic algorithms and natural adaptive systems.

Rule-set: a production rule system.

condition	message	action	strength
101#####	011010	110	[56]

System: A set of rules, receiving inputs from a message list and an input system; producing new messages and outputs to an output system.

Task: Reinforcement learning, system maps inputs to actions. Occasionally receives a reward.

Functioning:

1. Inputs put a message on the message list
2. Some rules match a message on the message list.
3. Those rules compete to respond to the message, based on their strength (and sometimes on the specificity of the rule).
4. The winner must pay (lose some strength) the sender of the message; it puts its message on the message list and its action is performed.
5. When reinforcement is received; it is given to the classifier posting the action. (Called a bucket-brigade algorithm.)

Adaptation: The set of rules is a population of a genetic algorithm. Crossover and mutation are used to generate new rules. Steady-state selection, with niching or similar to enforce different rules is used to propagate the effective rules. The strength of the rule is selected for.

Slide 10

Default Hierarchies: Rules which are specific (few #'s) are favoured in the conflict resolution and credit assignment schemes. Learning classifiers can adaptively create default hierarchies.

Default rule: A general rule (many #'s) which works in most but not all cases.

Exception rule: A specific rule which is satisfied by a subset of the messages matched by the default rule, and whose action corrects that of the default rule.

Slide 11

Using the population as a committee

Reference: Yao and Liu, 1996

Slide 12

A third way of using the entire population as the solution is to combine the outputs as a committee.

- Applied to training neural networks; but could be used for any learning system.
- Each member of the population represents a potential entire solution (as in the GABIL system).
- Implicit fitness sharing used to encourage different population members to perform correctly on different subsets of the example set.

Slide 13

- After GA is finished, entire population is used to produce output:
  - Majority voting
  - Voting weighted by fitness rank
  - Other methods tried

  Combined system was significantly better than the best member of the population on a majority of the real-world problems considered, but not on all.

Potentially, an inferior method of creating a committee machine due to the lack of independence of population members; but superior to throwing out all of the learning systems in the population but one.