Genetic Algorithms

Idea: learning based on simulated evolution

- Maintain population of competing hypotheses
- Evolve successor generations by recombining most fit, replacing least fit
- Terminate when sufficiently fit hypothesis obtained

Motivation:

- Biological evolution is robust, successful method
- Appropriate for complex hypotheses with many interacting parts
- Easily parallelized

Copyright Tom Mitchell, 25 October 1994

Typical Genenetic Algorithm

GA(Fitness, Acceptable-Fitness, p, r, m)

p: the number of hypotheses to be included in the population.

r: the fraction of the population to be replaced at each step.

m: the mutation rate.

Initialize population: P $\leftarrow$ Generate *p* hypotheses
Evaluate: for each h in P, compute Fitness(h)

While [MAX_hFitness(h) < Acceptable-Fitness] do

1. *Select:* $P_s \leftarrow$ the $(1-r)p$ members of P according to their fitness
2. *Crossover:* Randomly select $r \times p$ pairs of hypotheses from P_s, favoring those with highest fitness. For each pair, h_1, h_2, produce offspring by applying recombination operators. Add each offspring to P_s.
3. *Mutate:* For m percent of the members of P_s, flip one randomly selected bit in its representation.
4. *Update:* P $\leftarrow$ P_s
5. *Evaluate:* for each h in P, compute Fitness(h)

Return the hypothesis h from P that has the highest fitness.

Copyright Tom Mitchell, 25 October 1994

Fitness Determines Probability of Survival

Probability of selecting hypothesis h from current population:

$$\frac{Fitness(h_i)}{\sum_{j=1}^{p} Fitness(h_j)}$$

- Increases with fitness of h_i
- Decreases with fitness of its competitors

Copyright Tom Mitchell, 25 October 1994

Encoding Hypotheses as Bitstrings

Encoding a condition as a string:

Attribute a1 has possible values v1,...,v8

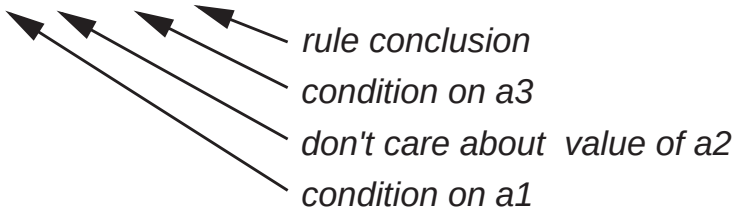
0 011 encodes "a1=v3"

1 10# encodes "a1 does NOT equal v4 or v5"

Encoding a rule as a string:

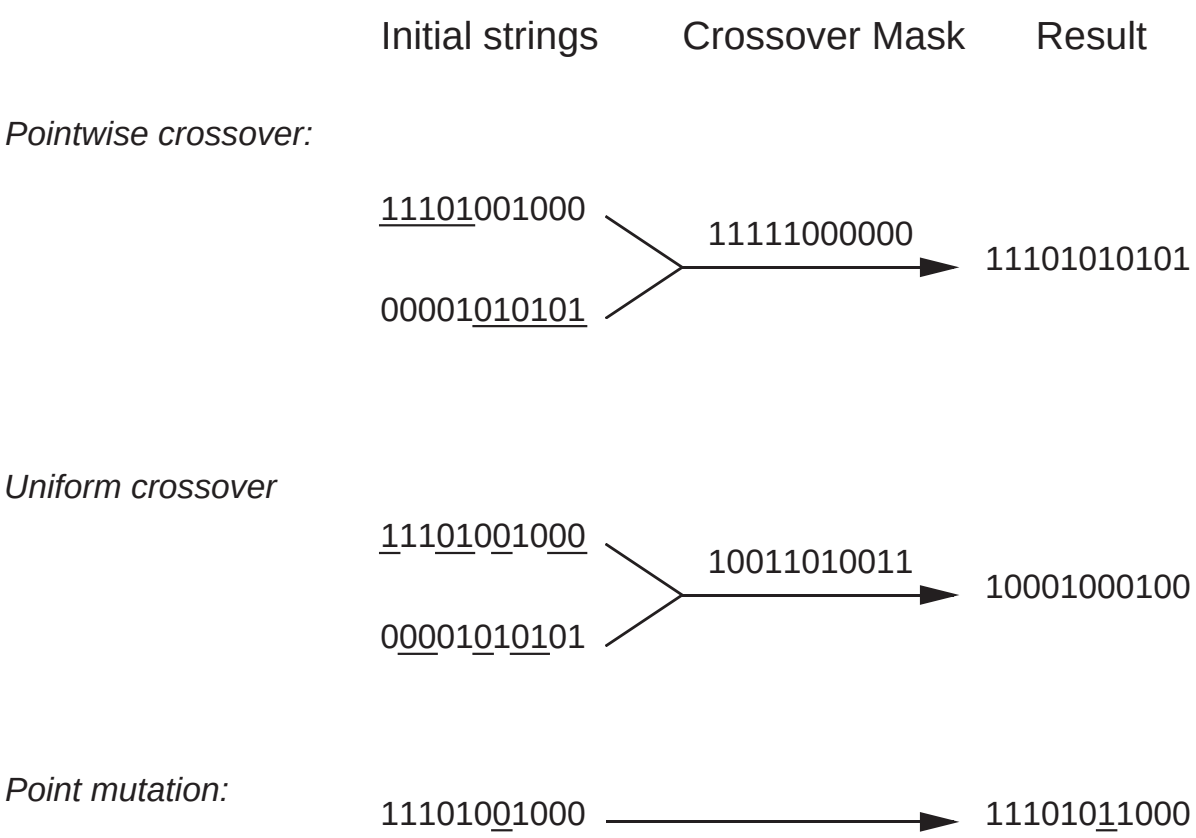
Assume rule can use 8-valued attritubte a1, and boolean attributes a2, a3
Assume rule concludes one of four conclusions, c1,...,c4

1011 ## 01 11 encodes "IF a1 does not equal v3, AND a3=v1, THEN c3"



Copyright Tom Mitchell, 25 October 1994

Common Recombination Operators



Copyright Tom Mitchell, 25 October 1994

An Example: COGIN

Task: Learn disjunctive set of rules

Approach:

- Population of bitsting/rules.
- Total population = disjunctive hypothesis
- Generation of successors is random
- Replacement within population is by competition, and considers total coverage of population

Copyright Tom Mitchell, 25 October 1994

COGIN Fitness Function

$$Fitness(R) = Info(R) + Lex(R)$$

$$Info(R) = Ent(S) - \frac{|Matched| Ent(Matched) + |Unmatched| Ent(Unmatched)}{|S|}$$

$$Lex(R) = \lceil \frac{N}{i} \rceil - \lceil \frac{Misclass(R)}{i} \rceil$$

Copyright Tom Mitchell, 25 October 1994

Generation of Successors

- Generate candidates: Old rules plus recombinations of all pairs.
- Recombination rule is randomly selected pointwise crossover
- Evaluate fitness of each rule
- Loop: Take the best, and subtract examples it classifies correctly

Copyright Tom Mitchell, 25 October 1994

More on GOGIN

See the following paper for the additional slides on COGIN

Green and Smith, "Competition Based Induction of Decision Models from Examples", *Machine Learning*, 13, Kluwer Academic Press, pp. 229- 257, November 1993.

Various Slides on Genetic Programming

sorry, I don’t have these online yet.

Copyright Tom Mitchell, 25 October 1994