COMP9417 Machine Learning and Data Mining

Note: Classifier Learning with Genetic Algorithms

Last revision: Thu May 3 2018

Introduction

This note gives an introduction to using genetic algorithms as a machine learning approach to learning classification rules. It will enable you to:

- outline the framework of evolutionary computation
- reproduce the prototypical genetic algorithm for machine learning
- design representations for rule learning by a genetic algorithm
- describe genetic algorithm operators such as mutation and crossov

[Recommended reading: "Machine Learning", T. Mitchell, Chapter 9]

Evolutionary Computation
What is this ?

- Computational procedures patterned after biological evolution
- Search method that *probabilistically* applies operators to set of points in the search space
- Can be viewed as form of *stochastic optimization*
  - aim to find approximate solutions to difficult optimization problems

Modelled (very loosely) on biological evolution:

- Lamarck and others:
  - Species "transmute" over time
- Darwin and Wallace:
  - Consistent, heritable variation among individuals in population
  - Natural selection of the "fittest"
- Mendel and genetics:
  - A mechanism for inheriting traits
  - mapping: genotype $\rightarrow$ phenotype

A Genetic Algorithm for Machine Learning
This algorithn captures the key features of a Genetic Algorithm (GA) method for learning classification rules. Note: several parameters are necessary; these must be supplied by the user.

GA($Fitness, Fitness\_threshold, p, r, m$)
Initialize: $P \leftarrow p$ random hypotheses
Evaluate: for each h in P, compute $Fitness(h)$
While $[\max_h Fitness(h)] < Fitness\_threshold$ Do

1. *Select:* Probabilistically select $(1-r)p$ members of P to add to P_S.
   $\Pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^{p} Fitness(h_j)}$
2. *Crossover:* Probabilistically select $\frac{r \cdot p}{2}$ pairs of hypotheses from P.
   For each pair, $\langle h_1, h_2 \rangle$, produce two offspring by
   applying the Crossover operator.
   Add all offspring to P_s.
3. *Mutate:* Invert a randomly selected bit in $m \cdot p$ random
   members of P_s
4. *Update:* $P \leftarrow P_s$
5. *Evaluate:* for each h in P, compute $Fitness(h)$

Return hypothesis from P with highest fitness.

How to represent hypotheses in a GA learner ? Basically, we have "chromosomes" encoded as *bitstrings* that represent attribute-value and class-value conditions as they would appear in rules. For example, we can represent:

$$(Outlook = Overcast \vee Rain) \wedge (Wind = Strong)$$

by

$$\begin{array}{cc} Outlook & Wind \\ 011 & 10 \end{array}$$

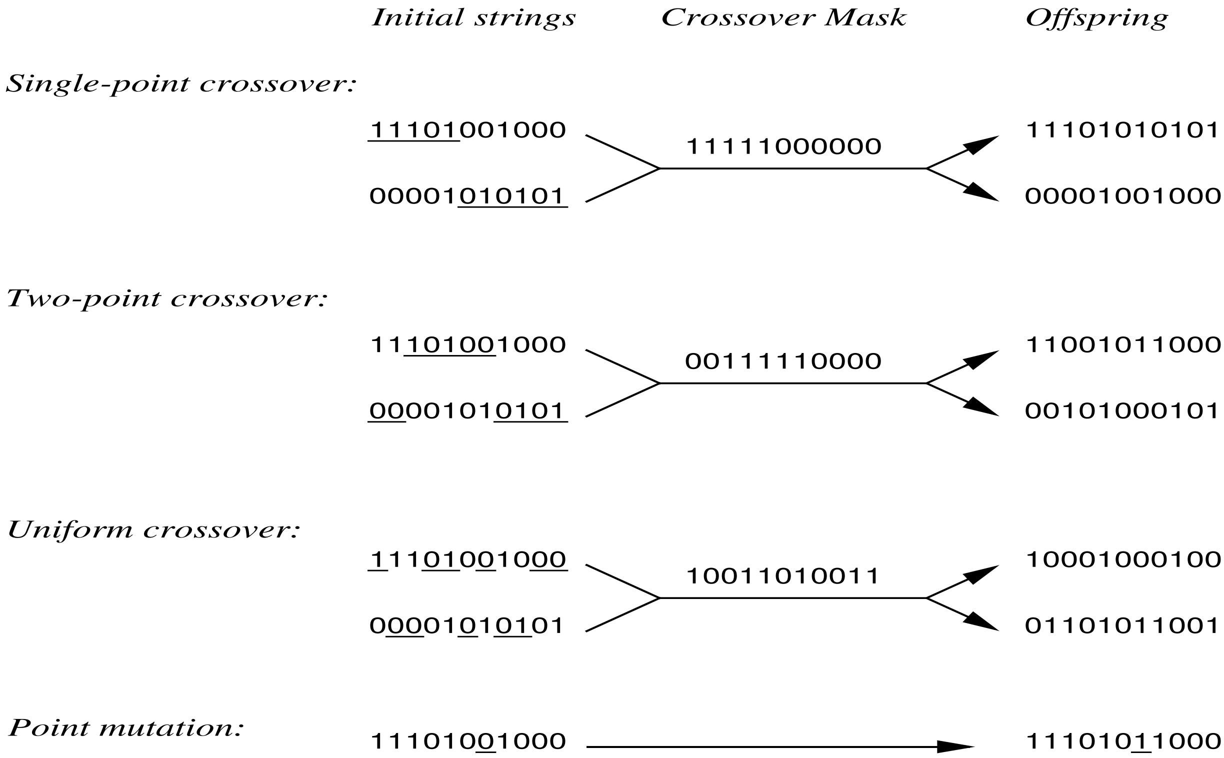
And

$$\text{IF} \quad Wind = Strong \quad \text{THEN} \quad PlayTennis = yes$$

by

$$\begin{array}{ccc} Outlook & Wind & PlayTennis \\ 111 & 10 & 10 \end{array}$$

Learning in a GA uses operators on bitstrings These are "biologically-inspired", but it is better to consider them as operators for randomly changing the components of hypotheses. In this diagram are shown versions of typical GA operators.



Mutation – new version of single parent
Crossover – two new offspring from two parents
Parameters for operators chosen randomly at each application
Point mutation: single bit chosen at random and flipped
Single-point crossover: n – number of bits contributed by first parent
Two-point crossover: n_0, n_1 – number of bits contributed by second, then first parent
Uniform crossover: non-contiguous bits chosen at random define contribution by each parent
Many variations of these operators are possible, and domain-specific versions can be designed.

Selecting hypotheses How does the GA measure *fitness* of hypotheses ? If hypotheses are rules:

- classification accuracy on data set, or
- accuracy combined with other factors, such as rule complexity

Fitness proportionate selection:

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

Also called *roulette wheel selection.*
Interpretation: select a hypothesis h on the basis of its fitness relative to the combined fitness of the population.

Other strategies may be better than fitness proportionate selection (can lead to "crowding" – many copies of similar individuals).
Tournament selection:

- Pick h_1, h_2 at random with uniform probability
- With probability p, select the more fit.

Rank selection:

- Sort all hypotheses by fitness
- Probability of selection is proportional to rank

Summary of GA rule learning In contrast to other classifier learning methods, the search for a successful hypothesis can be characterised as a randomized, parallel, hill-climbing search through the space of hypotheses, e.g., bitstring encodings of classification rules. It is best viewed as a stochastic approximation to optimizing the fitness function, which is based on minimizing error. For further details, refer to chapter 9 of Mitchell's textbook.

Acknowledgement: Material derived from slides for the book *Machine Learning*, Tom M. Mitchell, McGraw-Hill, 1997.