Genetic Algorithms

Genetic algorithms are used to generate successor hypotheses by repeatedly mutating and recombining parts of the best currently known hypotheses. At each step, a collection of hypotheses called the current population is updated by replacing some fraction of the population by offspring of the most fit current hypothesis.

This is advantageous because

1. Evolution is a successful, robust method
2. GAs can search a space of hypotheses containing complex interacting parts, where the impact of each part on overall fitness may be difficult to model
3. GA models can be easily parallelized and can take advantage of the decreasing costs of computer hardware.

The best hypothesis is defined by a hypothesis’ fitness. The algorithm operates by iteratively updating a pool of hypotheses called the population. Each iteration evaluates the fitness of each hypothesis and selects the most fit individuals to the next iteration, and some of them can be used as a basis to create new hypotheses.

The probability of selecting a hypothesis is given by

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

Representation

Hypotheses in GA are represented as bit strings, which can be very complex. The if-then rules can be encoded as bit strings, by seeing the number of values an attribute can take and assigning the attributes the required bits to characterize the values. The value 1 denotes that an attribute can take up that particular value.

Like this, several bit strings can be concatenated.

Another way to represent them is using integers. Another is a permutation representation.

Population

The population is a subset of solutions in the current generation. It is usually defined as population size, chromosome size.

Random initialisation is when the population is initialised with random solutions. Heuristic initialisation is when the population is initialised using a known heuristic.

Genetic Operators

Crossover

The crossover operator produces two new offspring from two parent strings, by copying selected bits from each parent. The crossover mask determines which parent contributes to which bit. A mask of 1111100000 indicates that the first 5 bits come from parent 1 and the rest from parent 2.

Types of crossovers

1. Single point : The crossover mask begins with n contiguous 1s followed by the necessary number of 0s
2. Two point : The mask begins with n zeroes, followed by m 1s, followed by the necessary number of 0s
3. Uniform : The mask combines bits sampled randomly from both the parents

Mutation

Mutation flips a single bit at random from the parent, creating a new hypothesis. This is usually applied after a crossover. Random resetting can also be done for integer representations where a random value from the set of permissible values is assigned. Swap mutation allows two positions' values to be swapped.

Fitness Function and Selection

The fitness function defines the criteria to rank hypotheses. It usually scores the classification accuracy of a rule over a set of training examples. Selection of a hypothesis can be done through proportionate selection as described above, or by tournament selection, where two hypotheses are chosen at random and with a predefined probability, the more fit of this is chosen. Rank selection can also be done where the probability of a hypothesis being chosen depends on the rank. Random selection can also be done.

Termination

The termination is reached when

1. No improvement in population fitness
2. When absolute number of generations are reached
3. When objective function has reached a certain threshold

Limitations

1. GAs are not suited for all problems, especially for problems that are simple
2. Fitness value is computationally expensive
3. No guarantee on optimality
4. Convergence might not happen

Some basic terminologies

1. Population : Subset of all possible solutions for the given problem
2. Chromosomes : One of the solutions (hypothesis)
3. Gene : One element position for the chromosome
4. Allele : Value a gene takes for a particular chromosome
5. Genotype : Population in the computation space
6. Phenotype : Population in the actual real world solution

A generic pseudo code

```
1. Initialise population
2. Evaluate the fitness of the population
3. While (not terminate) do
    a. Parent selection
    b. Do crossover with some probability p
    c. Do mutation with some probability q
    d. Calculate fitness
    e. Select survivors
    f. Find the best survivor
4. Return the best survivor
```