Machine Learning

Genetic Algorithm

Lecturer: Duc Dung Nguyen, PhD.
Contact: nddung@hcmut.edu.vn

Faculty of Computer Science and Engineering
Hochiminh city University of Technology

Contents

Introduction

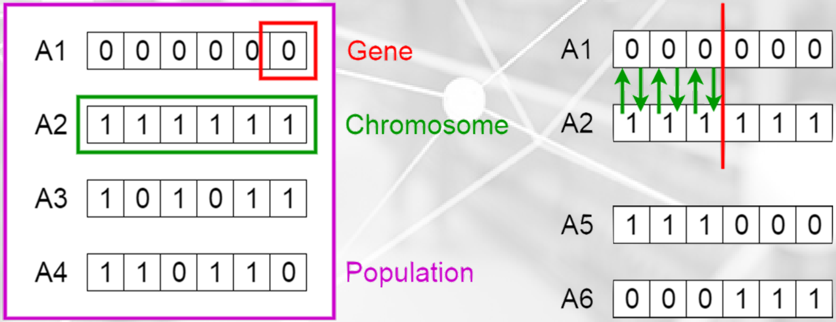
Motivation

- **Evolution** is known to be a successful, robust method of nature
- How do we **search** the **space of hypotheses** containing complex interacting parts, where the impact of each part on overall hypothesis is **hard to model**.
- Computer programs are evolved to **certain fitness criteria**.
- Evolutionary computation = Genetic algorithms + genetic programming

- **Learning** as **searching**
- Analogy to **biological evolution**
  - The best hypothesis is searched through **several generations** of hypotheses.
  - Next generation hypotheses are produced by **mutating** and **recombining** parts of the best current generation hypotheses
- It is not recommended to search from general-to-specific or from simple-to-complex hypotheses.

Genetic Algorithm

Genetic Algorithms



A Prototypical GA

- **Initialize population**: $P =$ randomly generated p hypotheses.
- **Evaluate fitness**: compute $Fitness(h)$, for each $h \in P$
- While $max_{h \in P} Fitness(h) < Fitness_{threshold}$ do:
  - Create new generation
  - Evaluate fitness

New Generation Creation

- Selection
- Crossover
- Mutation

- Selection:

Probabilistically select $(1-r)p$ hypotheses of P to add to the new generation.

The selection probability of a hypothesis

$$Pr(h_i) = \frac{Fitness(h_i)}{\sum_{h \in P} Fitness(h)}$$

New Generation Creation

- Crossover:
  - Probabilistically select $(r/2)p$ pairs of hypotheses from P according to $Pr(h)$
  - For each pair (h_1, h_2), produce two offsprings by applying a Crossover operator.
  - Add all offspring to the new generation.

New Generation Creation

- Mutation:
  - Choose m percent of the added hypotheses with **uniform distribution**.
  - For each, **invert** one **randomly selected bit** in its representation.

Hypotheses Representation

- A classification rule as a bit string:

$$\text{If} \quad expr(A_1) \wedge ... \wedge expr(A_i \wedge ... \wedge expr(A_n) \quad \text{Then} \quad C = c$$

Hypothesis Representation

- Example:

$$\text{If } \ Wind = Strong \ \ \text{Then } \ PlayTennis = Yes$$

$$\Downarrow$$

1	1	1	1	0	1	0

Hypothesis Representation

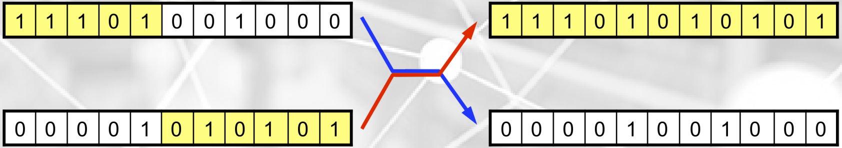
- A set of rules as concatenated bit strings:

1	1	1	1	0	1	0	1	0	0	0	1	0	1

Crossover Operator

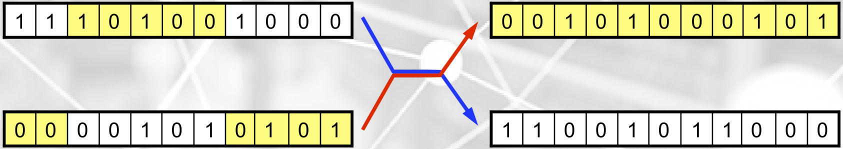
- Single-point
- Two-point
- Uniform

- Single-point



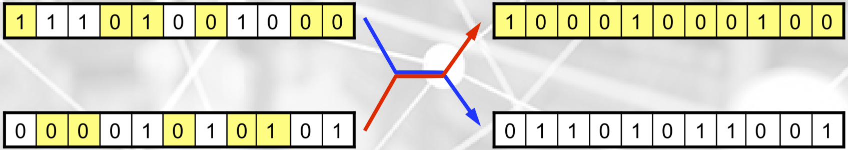
Crossover Operator

- Two-point

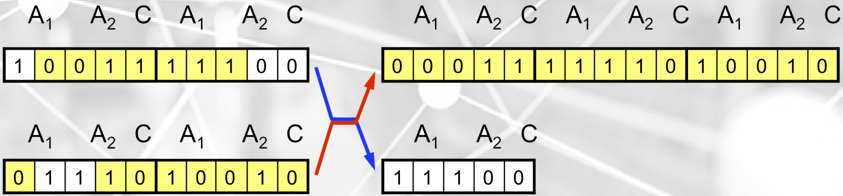


Crossover Operator

- Uniform



- Variable-length bit strings



Fitness Function

- Example:

$$Fitness(h) = (correct(h))^2$$

$correct(h)$ = percent of all training examples correctly classified by hypothesis h

Inductive bias?

- What is **inductive bias**?
- Where is inductive bias in GA?