Genetic Algorithms

- Learning as searching.

- Analogy to biological evolution: the best hypothesis is searched thru several generations of hypotheses.

- Next generation hypotheses are produced by mutating and recombining parts of the best current generation hypotheses.

  (Rather than search from general-to-specific or from simple-to-complex hypotheses)

GA Advantages

- Evolution is a successful, robust method for adaptation within biological systems.
- GAs can search spaces of hypotheses containing complex interacting parts.
- GAs are easily parallelized.

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 a new generation

  Evaluate fitness

New Generation Creation

- Selection
- Crossover
- Mutation

New Generation Creation

- Selection:

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

  The selection probability of a hypothesis $\Pr(h_i) = \dfrac{\text{Fitness}(h_i)}{\sum_{h \in P} \text{Fitness}(h)}$

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 offspring 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.

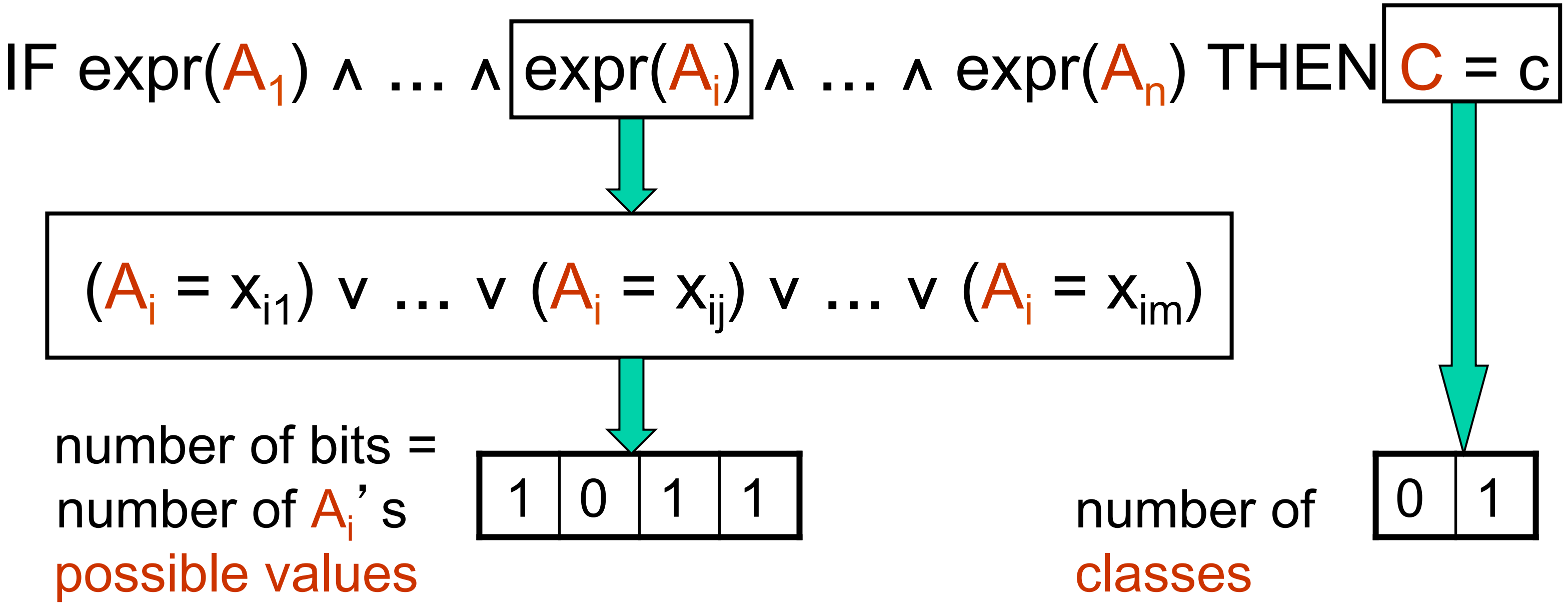
Hypothesis Representation

- A classification rule as a bit string:

  IF expr(A_1) ∧ … ∧ expr(A_i) ∧ … ∧ expr(A_n) THEN C = c

Hypothesis Representation

- A classification rule as a bit string:



IF $\mathrm{expr}(A_1) \wedge \ldots \wedge \boxed{\mathrm{expr}(A_i)} \wedge \ldots \wedge \mathrm{expr}(A_n)$ THEN $\boxed{C = c}$

$(A_i = x_{i1}) \vee \ldots \vee (A_i = x_{ij}) \vee \ldots \vee (A_i = x_{im})$

number of bits = number of A_i' s possible values

1	0	1	1

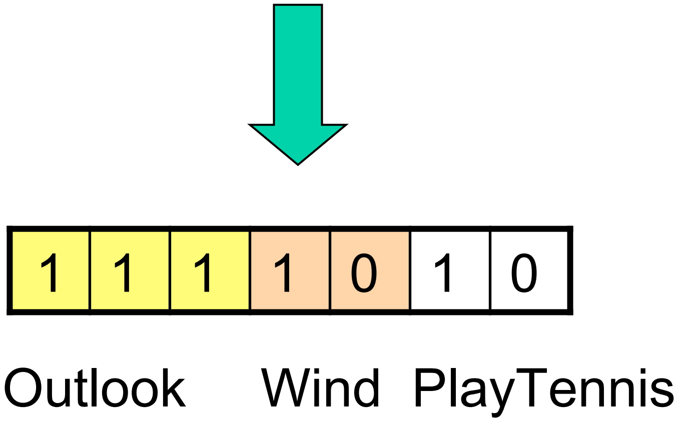
number of classes

0	1

Hypothesis Representation

- Example:

IF Wind = Strong THEN PlayTennis = Yes



1	1	1	1	0	1	0

Outlook Wind PlayTennis

Hypothesis Representation

- A set of rules as concatenated bit strings:

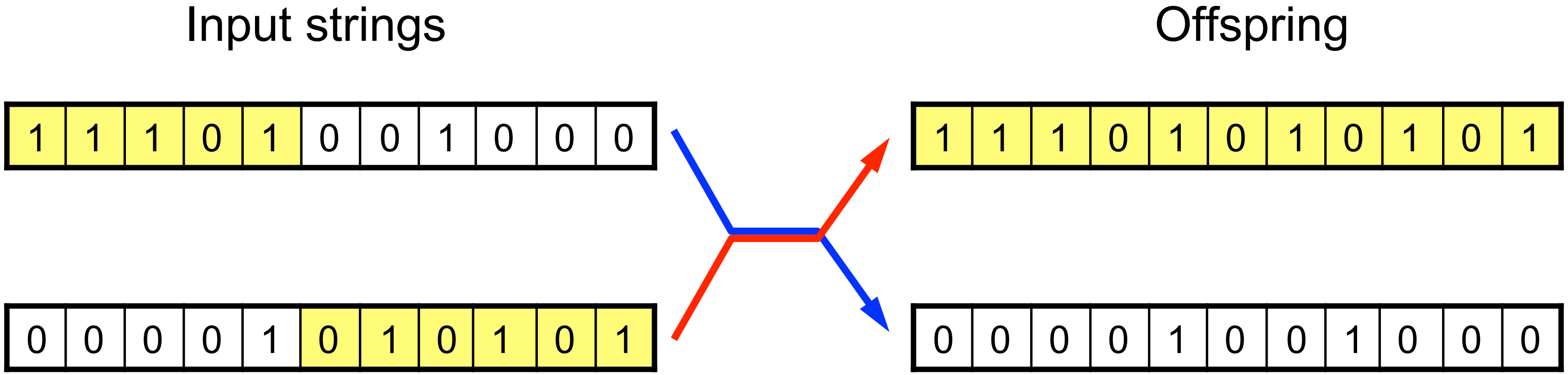
1	1	1	1	0	1	0	1	0	0	0	1	0	1
Rule 1							Rule 2						

Crossover Operator

- Single-point
- Two-point
- Uniform

Crossover Operator

- Single-point



Input strings

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

Offspring

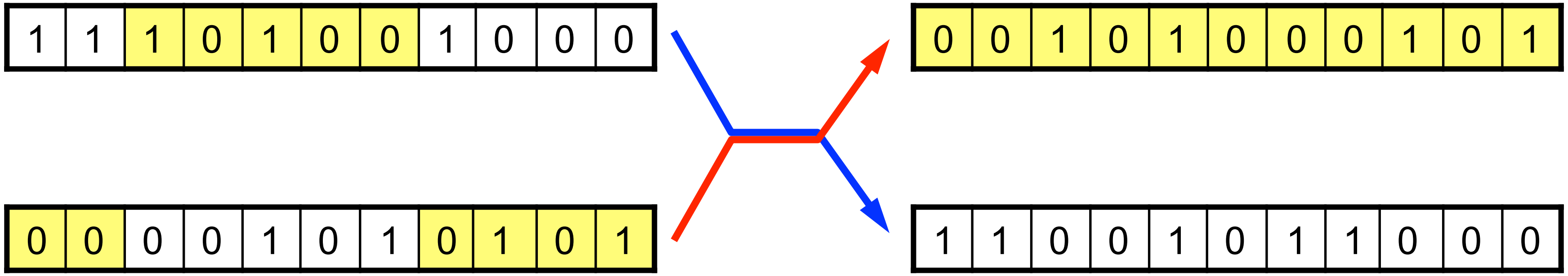
1	1	1	0	1	0	1	0	1	0	1
0	0	0	0	1	0	0	1	0	0	0

Crossover Operator

- Two-point

Input strings

Offspring

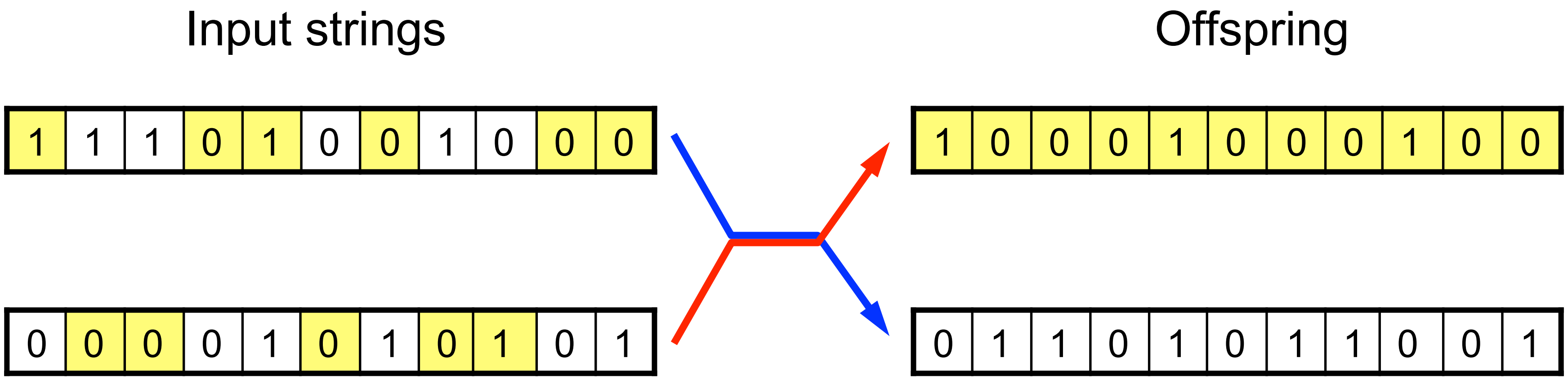


1	1	1	0	1	0	0	1	0	0	0
0	0	0	0	1	0	1	0	1	0	1

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

Crossover Operator

- Uniform



Input strings

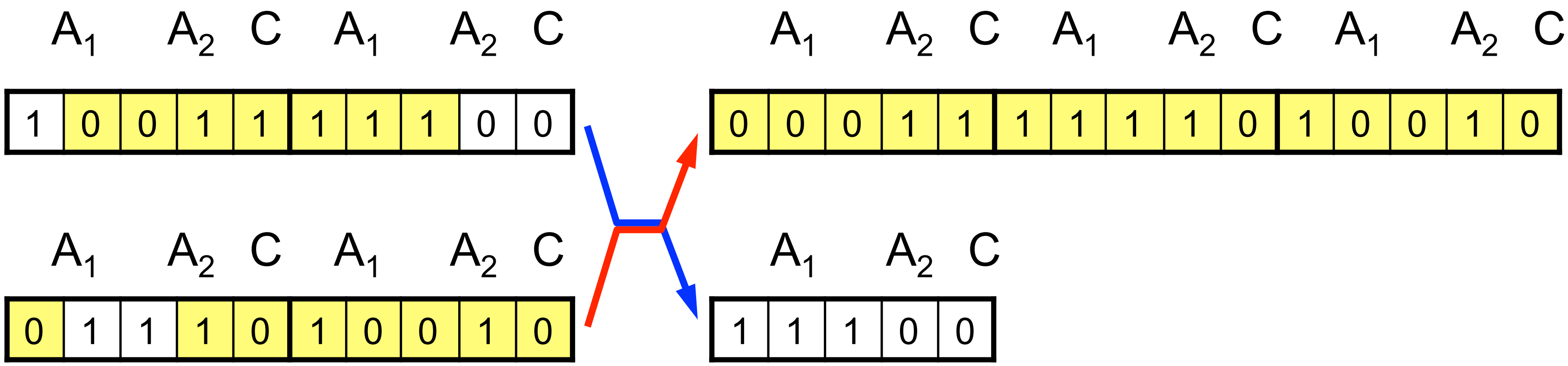
1	1	1	0	1	0	0	1	0	0	0
0	0	0	0	1	0	1	0	1	0	1

Offspring

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

Crossover Operator

- Variable-length bit strings



Input strings

A_1 A_2 C A_1 A_2 C

1 0 0 1 1 1 1 1 0 0

A_1 A_2 C A_1 A_2 C

0 1 1 1 0 1 0 0 1 0

Offspring

A_1 A_2 C A_1 A_2 C A_1 A_2 C

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

A_1 A_2 C

1 1 1 0 0

Fitness Function

- Example:

$$\text{Fitness}(h) = (\text{correct}(h))^2$$

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

Where is GA inductive bias?

Exercises

- In Mitchell’s ML (Chapter 9): 9.1, 9.2