Lecture 7: Genetic Algorithms

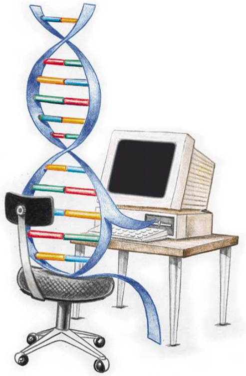
Cognitive Systems II - Machine Learning

SS 2005

Part II: Special Aspects of Concept Learning

Genetic Algorithms, Genetic Programming, Models of Evolution

Motivation



- learning methods is motivated by analogy to biological evolution

- rather than search from general-to-specific or from simple-to-complex, genetic algorithms 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 by offspring of the most fit current hypotheses

Motivation

- reasons for popularity
  - evolution is known to be a successful, robust method for adaption within biological systems
  - genetic algorithms can search spaces of hypotheses containing complex interacting parts, where the impact of each part on overall hypothesis fitness may be difficult to model
  - genetic algorithms are easily parallelized
- genetic programming $\approx$ entire computer programs are evolved to certain fitness criteria
- evolutionary computation $=$ genetic algorithms $+$ genetic programming

Genetic Algorithms

- **problem:** search a space of candidate hypotheses to identify the best hypothesis
- the best hypothesis is defined as the one that optimizes a predefined numerical measure, called **fitness**
  - e.g. if the task is to learn a strategy for playing chess, fitness could be defined as the number of games won by the individual when playing against other individuals in the current population
- **basic structure:**
  - iteratively updating a pool of hypotheses (**population**)
  - on each iteration
    - hypotheses are evaluated according to the **fitness function**
    - a new population is generated by selecting the most fit individuals
    - some are carried forward, others are used for creating new offspring individuals

Genetic Algorithms

GA($Fitness, Fitness\_threshold, p, r, m$)

Fitness: fitness function, Fitness_threshold: termination criterion,
p: number of hypotheses in the population, r: fraction to be replaced by crossover,
m: mutation rate

- Initialize population: $P \leftarrow$ Generate p hypotheses at random
- Evaluate: For each h in P, compute $Fitnes(h)$
- While $[\max_h Fitness(h)] < Fitness\_threshold$, Do
  1. **Select:** Probabilistically select $(1 - r) \cdot p$ members of P to add to P_S
  2. **Crossover:** Probabilistically select $\frac{r \cdot p}{2}$ pairs of hypotheses from P. For each pair $< h_1, h_2 >$ produce two offspring and add to P_S
  3. **Mutate:** Choose m percent of the members of P_S with uniform probability. For each, invert one randomly selected bit
  4. **Update:** $P \leftarrow P_S$
  5. **Evaluate:** for each $h \in P$, compute $Fitness(h)$
- Return the hypothesis from P that has the highest fitness.

Remarks

- as specified above, each population P contains p hypotheses
- $(1-r)\cdot p$ hypotheses
  - are selected and added to P_S without changing
  - the selection is probabilistically
  - the probability is given by $Pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^{p} Fitness(h_j)}$
- $\frac{r\cdot p}{2}$ pairs of hypotheses
  - are selected and added to P_S after applying the crossover operator
  - the selection is also probabilistically

$\Rightarrow$ $(1-r)\cdot p + 2\cdot\frac{r\cdot p}{2} = p$ where $r + (1-r) = 1$

Representing Hypotheses

- hypotheses are often representated as **bit strings** so that they can easily be modified by genetic operators
- representated hypotheses can be quite complex
- each attribute can be representated as a subtring with as many positions as there are possible values
- to obtain a fixed-length bit string, each attribute has to be considered, even in the most general case
  - $(Outlook = Overcast \vee Rain) \wedge (Wind = Strong)$
  - is representated as: $Outlook\ 011$, $Wind\ 10 \Rightarrow 01110$

Genetic Operators

- generation of successors is determined by a set of operators that recombine and mutate selected members of the current population
- operators correspond to idealized versions of the genetic operations found in biological evolution
- the two most common operators are **crossover** and **mutation**

Genetic Operators

	Initial strings	*Crossover Mask*	*Offspring*
Single-point crossover:	11101001000	11111000000	11101010101
	00001010101		00001001000
Two-point crossover:	11101001000	00111110000	11001011000
	00001010101		00101000101
Uniform crossover:	11101001000	10011010011	10001000100
	00001010101		01101011001
Point mutation:	11101001000		11101011000

Genetic Operators

- **Cossover:**
  - produces two new offspring from two parent strings by copying selected bits from each parent
  - bit at position i in each offspring is copied from the bit at position i in one of the two parents
  - choice which parent contributes bit i is determined by an additional string, called **cross-over mask**
    - **single-point crossover:** e.g. 11111000000
    - **two-point crossover:** e.g. 00111110000
    - **uniform crossover:** e.g. 01100110101
- **mutation:** produces bitwise random changes

Illustrative Example (GABIL)

- GABIL learns boolean concepts represented by a disjunctive set of propositional rules
- **Representation:**
  - each hypothesis is encoded as shown above
  - hypothesis space of rule preconditions consists of a conjunction of constraints on a fixed set of attributes
  - sets of rules are representates by concatenation
  - e.g. a_1, a_2 boolean attributes, c target attribute

    IF $a_1 = T \wedge a_2 = F$ THEN $c = T$;

    IF $a_2 = T$ THEN $c = F$

    $\Rightarrow 10\ 01\ 1\ 11\ 10\ 0$

Illustrative Example (GABIL)

- **Genetic Operators:**
  - uses standard mutation operator
  - crossover operator is a two-point crossover to manage variable-length rules

  - **Fitness function:**
    - $Fitness(h) = (correct(h))^2$
    - based on classification accuracy where $correct(h)$ is the percent of all training examples correctly classified by hypothesis h

Hypothesis Space Search

- method is quite different from other methods presented so far
- neither general-to-specific nor simple-to-complex search is performed
- genetic algorithms can move **very abruptly**, replacing a parent hypothesis by an offspring which is radically different
- so this method is less likely to fall into some local minimum
- practical difficulty: **crowding**
  - some individuals that fit better than others reproduce quickly, so that copies and very similar offspring take over a large fraction of the population

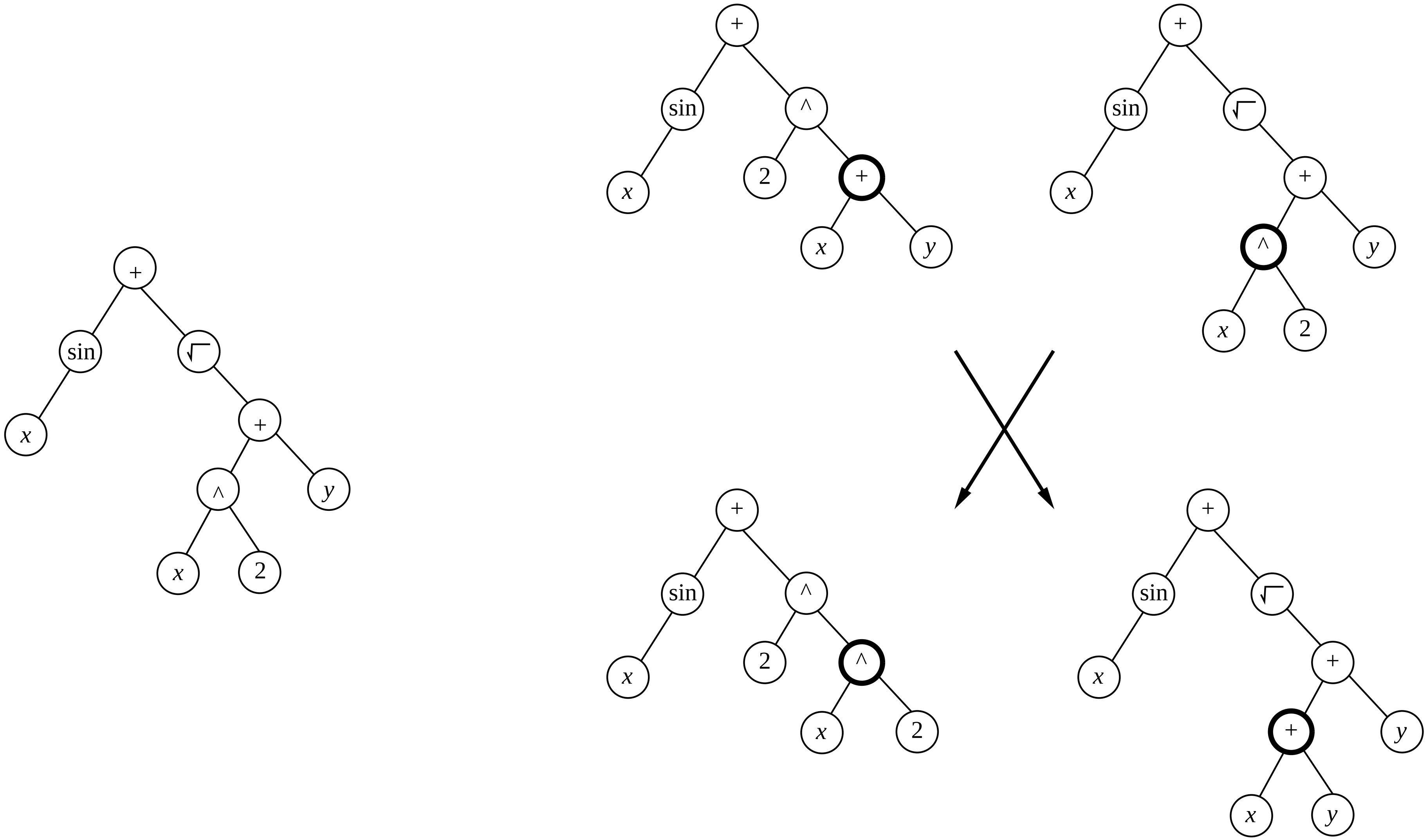
  $\Rightarrow$ reduced diversity of population

  $\Rightarrow$ slower progress of the genetic algorihms

Genetic Programming

- individuals in the evolving population are computer programs rather than bit strings
- has shown good results, despite vast H
- **representing programs**
  - typical representations correspond to parse trees
    - each function call is a node
    - arguments are the descendants
  - fitness is determined by executing the programm on the training data
  - crossover are performed by replacing a randomly chosen subtree between parents

Genetic Programming



Models of Evolution and Learning

- **observations:**
  - individual organisms learn to adapt significantly during their lifetime
  - biological and social processes allow a species to adapt over a time frame of many generations
- interesting question: What is the relationship between learning during lifetime of a single individual and species-level learning afforded by evolution?

Models of Evolution and Learning

- **Lamarckian Evolution:**
  - proposition that evolution over many generations was directly influenced by the experiences of individual organisms during their lifetime
  - direct influence of the genetic makeup of the offspring
  - completely contradicted by science
  - Lamarckian processes can sometimes improve the effectiveness of genetic algorithms
- **Baldwin Effect:**
  - a species in a changing environment underlies evolutionary pressure that favors individuals with the ability to learn
  - such individuals perform a small local search to maximize their fitness
  - additionally, such individuals rely less on genetic code
  - thus, they support a more diverse gene pool, relying on individual learning to overcome “missing” or “not quite well” traits

  $\Rightarrow$ indirect influence of evolutionary adaption for the entire population

Summary

- method for concept learning based on simulated evolution
- evolution of populations is simulated by taking the most fit individuals over to a new generation
- some indiviuals remain unchanged, others are the base for genetic operator application
- hypotheses are commonly representated as bitstrings
- search through the hypothesis space cannot be characterized, because hypotheses are created by crossover and mutation operators that allow radical changes between successive generations
- hence, convergence is not guaranteed