Genetic Algorithms

CS 550: Machine Learning

Genetic Algorithm

- It conducts a randomized, parallel, and hill-climbing search for hypotheses (solutions) that optimize a predefined fitness function

- This search is based on an analogy to biological evolution
  - It maintains a diverse population of competing hypotheses (they can be considered as individuals who are fighting for survival within a larger population)
  - It generates successor hypotheses by repeatedly mutating and recombining parts of the best currently known hypotheses (among these individuals only the fittest ones will reproduce and survive)

Genetic Algorithm

- The genetic algorithm searches for global maxima/minima
- However, it does not guarantee to find one
- Compare this search with the gradient descent algorithm

Genetic Algorithm

initialization: initialize population to contain p individuals

repeat

selection: probabilistically (according to the fitness function) select (1 – r) p individuals and add them to the next generation

crossover: probabilistically (according to the fitness function) select r.p / 2 pairs from the population, crossover each pair, and add the two offspring to the next generation

mutation: choose m.p individuals to mutate

until **stopping criterion** is satisfied

Create a new generation

Hypothesis Representation

- Binary encoding is commonly used
  - Other encoding schemes are also possible
  - Other alphabets can also be used
- Easier to represent discrete features
- However, it could be complicated for continuous features
  - Number of bits dedicated to a particular feature depends on upper and lower bounds of the feature values and the precision
  - Insufficient resolution may result in loss in precision

$$no_{bits} = \left\lceil \log_2 \frac{x_{\max} - x_{\min} + 1}{\Delta x} \right\rceil \qquad \Delta x = \frac{x_{\max} - x_{\min} + 1}{2^{no_{bits}}}$$

Example: Binary Encoding

- 4-class classification problem with the following features
  - Gender (*female or male*) ← 2 distinct values, 1 bit
  - Age (*0-127 years, intervals of 1 year*) ← 128 distinct values, 7 bits
  - Height (*0.50 – 2.50 meters, intervals of 0.5 cms*) ← 401 distinct values, 9 bits
- Now consider representing a person
  - Female ← 1
  - 41 years old ← 0 1 0 1 0 0 1
  - 1.76 m tall ← 0 1 1 1 1 1 1 0 0 (=252, 0.50 + 252 * 0.005 = 1.76)
  - 3rd class ← 0 0 1 0 (if 4 bits are used) OR 1 0 (if 2 bits are used)

$$\underbrace{1}_{\textit{Female}}\ \underbrace{0\ 1\ 0\ 1\ 0\ 0\ 1}_{\textit{Age}}\ \underbrace{0\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0}_{\textit{Height}}\ \underbrace{0\ 0\ 1\ 0}_{\textit{Class}}$$

One should be careful about not to generate invalid binary encodings after crossover and mutation operations

Example: Binary Encoding

- What about to encode the learning models?
  - Could be hard to encode some learning models
  - Easier to encode if-and-then rules
- Decide whether or not to play tennis depending on
  - Outlook (sunny, cloudy, rainy)
  - Wind (strong, weak)

$$\underbrace{0\ 1\ 1}_{\text{Outlook}}\ \underbrace{1\ 0}_{\text{Wind}}\ \underbrace{0\ 1}_{\text{Play?}}$$

if (outlook = cloudy or rainy) and (wind = strong) then play = NO

1 1 1 0 1 1 0

if (wind = weak) then play = YES

One should be careful about not to generate invalid binary encodings after crossover and mutation operations

Genetic Operators: Crossover and Mutation

Single point crossover

Parents	Mask	Offspring
1 1 0 0 | 0 0 1	1 1 1 1 0 0 0	1 1 0 0 1 1 0
0 0 0 1 | 1 1 0		0 0 0 1 0 0 1

Two-point crossover

Parents	Mask	Offspring
1 1 | 0 0 0 | 0 1	1 1 0 0 0 1 1	1 1 0 1 1 0 1
0 0 | 0 1 1 | 1 0		0 0 0 0 0 1 0

Multipoint crossover

Parents	Mask	Offspring
1 1 0 0 0 0 1	1 0 1 1 0 1 0	1 0 0 0 1 0 0
0 0 0 1 1 1 0		0 1 0 1 0 1 1

Point mutation

1 1 0 <u>0</u> 0 0 1 → 1 1 0 <u>1</u> 0 0 1

Be careful about not to generate invalid encodings after crossover and mutation!!!

Fitness Function and Selection

- To create a new population, individuals are probabilistically selected according to their fitness values
- Need to define a fitness function based on the criteria you want to optimize
  - Classification accuracy
  - Complexity or generality of the rules
  - Classification cost together with feature extraction cost
- Invalid encodings can also be penalized through the fitness function

Selection Techniques

Fitness proportionate selection (roulette wheel selection)

- Select individuals by a probability that is directly proportional to the raw fitness value

example :

$$\Pr(h_i) = \frac{fitness(h_i)}{\sum_j fitness(h_j)}$$

One practical difficulty is the problem of **CROWDING.** It occurs when some individuals are more highly fit than the others especially in the first generations. In this case, they quickly reproduce and very similar individuals take over a large fraction of the population. This reduces the diversity of population.

Selection Techniques

Rank selection

- Sort the hypotheses according to their fitness values
- Then the selection probability is proportional to the ranks (not the raw fitness values)
- It is useful to preserve the diversity better, alleviating the crowding problem

example :

$$\Pr(h_i) = \frac{1/rank(h_i)}{\sum_j 1/rank(h_j)}$$

Selection Techniques

Tournament selection

- Randomly select two hypotheses
- Then select the fittest one among these two
- Repeat these two steps until you select what you need
- Similarly, it is useful to preserve the diversity better, alleviating the crowding problem

Crowding

Possible techniques to alleviate the crowding problem

- Alter the selection method: use rank selection or tournament selection
- Use fitness sharing strategy: reduce the fitness of a hypothesis by the presence of other similar hypotheses in the population
- Restrict the kinds of hypotheses allowed to recombine for forming offspring

Initialization and Stopping

Initialization

- Goal is to select an initial population that has both quality and diversity

Stopping criterion examples include to stop when

- A specified number of iterations is reached
- Genetic diversity between the hypotheses is small
- No or marginal improvement is achieved from the current generation to the next
- The fitness value of the fittest hypothesis reaches the targeted goal

Variants to Create Next Generation

- Fittest individuals may survive as unchanged (the algorithm given in Page 4 uses this strategy)
- Or replace an entire population at a time, no individuals survive (r = 1 in the given algorithm)
- Select two parents, crossover them, but eliminate only one of them by replacing it with the fittest offspring
- Only crossover “dissimilar” parents

Genetic Programming

- It is a variant of genetic algorithms
- Hypotheses are computer programs rather than bit strings (tree representation could be used)
- The aim is to find a computer program that performs well in a predefined task with respect to a fitness function
- Crossover and mutation operations should be defined to be applied on programs