BITS F464: Machine Learning

28 Genetic Algorithms

Dr. Kamlesh Tiwari
Assistant Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

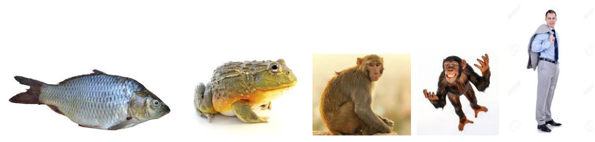
March 31, 2021 ONLINE (Campus @ BITS-Pilani Jan-May 2021)

http://ktiwari.in/ml

Genetic Algorithms

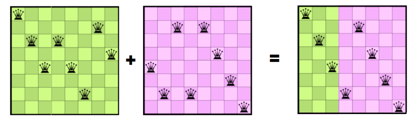
Evolution

Recall Darwin's theory of evolution: "*Survival of the fittest*"



Is it true? Let's formulate and try..

- 8-queens problem



Note*: Images taken from various sources on Internet

Genetic Algorithms

Learning approach of **genetic algorithms** is based on simulated evolution (appeared in 1975)

- Search for an appropriate hypothesis begins with a population of initial hypotheses
  - Members of the current population give rise to the population of next generation using **selection**, **mutation** and **crossover**
  - Hypotheses are evaluated using some `Fitness` measure
  - Most fit hypotheses act as a seeds for producing next generation
- Applied a variety of learning tasks and optimization problems (like robot control and learning parameters for ANN)
- GA is an generate-and-test approach. Search can be very complex, (may involve hill climbing, simulated annealing ...)
- Search can move abruptly. Crowding can happen

Without **guarantee**, GA often finds an object of high fitness

Hypotheses as fixed length bit strings

Hypotheses are represented as chromosomes (fixed length bit strings)

- Consider an attribute *Outlook*, which takes value from the set {Sunny, Overcast, Rain}
- Let string 010 means *Outlook*=Overcast
- Let string 011 means *Outlook*=Overcast ∨ Rain
- Consider an attribute *Wind* taking values from {Strong, Weak}
- The a rule such as (*Outlook*=Overcast ∨ Rain) ∧ (*Wind*=Strong) is written as 01110
- Post condition *playTennis*=Yes is incorporated as 0111010 assuming attribute *playTennis* can take any value from {Yes, No} However, 011101 also be used (single bit for boolean attribute)

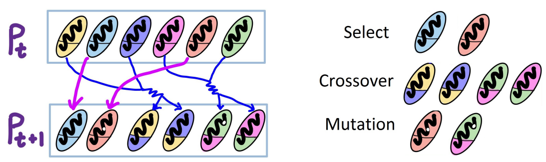
This representation contains a substring for each attribute, **even if the attribute is not constrained by the rule preconditions**

This yields a fixed length bit-string representation for rules

Genetic Algorithms

Algorithm 1: GA (`Fitness`, F_{th}, p, r, m)

P ← generate p states at random
while $max($`Fitness`(h_1), `Fitness`(h_2), ..., `Fitness`$(h_p)) < F_{th}$ **do**
  Select: $(1-r)p$ members of P
  Crossover: on $(r \times p)/2$ pairs to produce two offspring
  Mutation: invert, at random, a bit, of m fraction of population
return state h_i having maximum `Fitness`(h_i)



Fitness Function

- `Fitness` function is typically a heuristic, that never overestimates.
- It function defines a criterion for ranking hypotheses
- Ranking is used to probabilistically select hypotheses for inclusion in the population of next generation
- To learn classification, the fitness function typically is accuracy
- Although, one can use complexity or generality of the rule
- Or overall performance of the resulting procedure rather than performance of individual rules

Approaches For GA Selection

1. **Fitness proportionate**. Probability of selecting a hypothesis in next generation is

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

2. **Tournament selection**. randomly pick two states and then with some predefined probability p the more fit of these two is then selected, and with probability $(1-p)$ the less fit hypothesis is selected
3. **Rank selection**. states are sorted by `Fitness` and the probability of selection of a hypothesis is proportional to its rank in this sorted list, rather than its `Fitness`
4. **Elitist Model**. select a small proportion of the fittest candidates of current population as it is intact into the next generation

Genetic Operators (crossover and mutation)

	Initial String	Crossover Mask	Offspring
Single Point **Crossover**	11101001000 00001010101	11111000000	11101010101 00001001000
Two Point **Crossover**	11101001000 00001010101	00111110000	11001011000 00101000101
Uniform **Crossover**	11101001000 00001010101	10011010011	10001000100 01101011001
Point **Mutation**	11101001000	→	11101011000

Does GA works?

Can we mathematically characterize the evolution?

- **Schema:** string of 0, 1 or $*$ like 0*1 denoting set $\{001, 011\}$
- String 1011 is **representative** of 2^4 schema
- Consider p_t, population at time t, if fitness of individual h be $f(h)$, what is average fitness of the population $\bar{f}(t) = \frac{1}{n}\sum f(h)$
- Let population contains $m(s,t)$ instances of a schema s
- Consider *selection*, let $h \in s \cap p_t$ denotes that h is representative of s and it is present in the population at time t
- Let $\hat{u}(s,t)$ be average fitness of instances of s at time t

$$\hat{u}(s,t) = \frac{\sum_{h \in s \cap p_t} f(h)}{m(s,t)}$$

- We know $Pr(h) = f(h)/(\sum f(h)) = f(h)/(n\bar{f}(t))$

Does GA works? (contd..)

- Probability that we will select a representative of schema s is

$$\begin{aligned} Pr(h \in s) &= \sum_{h \in s \cap p_t} Pr(h) = \sum_{h \in s \cap p_t} f(h)/(n\bar{f}(t)) = \frac{1}{n\bar{f}(t)} \sum_{h \in s \cap p_t} f(h) \\ &= \frac{\hat{u}(s,t)}{n\bar{f}(t)} m(s,t) \end{aligned}$$

- Expected number of instances of s resulting from the n independent selection steps that create the entire new generation is just n times this probability. Therefore,

$$E[m(s,t+1)] = \frac{\hat{u}(s,t)}{\bar{f}(t)} m(s,t)$$

Does GA works? (contd..)

$$E[m(s,t+1)] = \frac{\hat{u}(s,t)}{\bar{f}(t)} m(s,t)$$

Expected number of representative instances of a schema s in the generation at time $t+1$ is

1. Proportional to the average fitness $\hat{u}(s,t)$ of instances of this schema at time t , and
2. Inversely proportional to the average fitness $\bar{f}(t)$ of all members of the population at time t

Thus, we can expect schema with above average fitness to be represented with increasing frequency on successive generations

Does GA works? (contd..)

Also consider negative effects of single point crossover and mutation

- Let p_c represents the probability of **crossover** on an individual. $d(s)$ be the distance between left most and right most defined bit of s and l be the length of individual bit string
- Let p_m represents the probability of **mutation** on an individual and $o(s)$ be number of defined bits in s

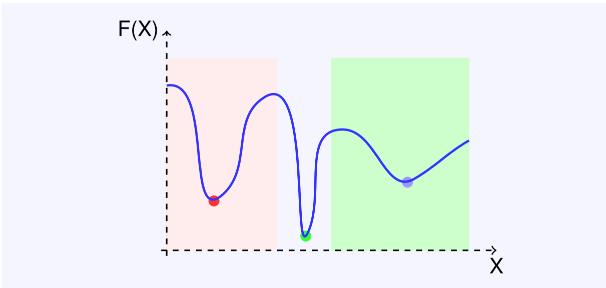
Full schema theorem thus provides a lower bound on the expected frequency of schema s, as follows

$$E[m(s,t+1)] \geq \frac{\hat{u}(s,t)}{\bar{f}(t)} m(s,t)\left(1 - p_c \frac{d(s)}{l-1}\right)(1-p_m)^{o(s)}$$

Similar expression. influence of more fit schema will tend to grow

Optimization in GA

GA can be viewed as a general optimization method.



- Any traditional algorithms (such as gradient descent) with initial guess in pink or violet region leads to local minima
- GA would lead to global one
- Hybrid approach start with GS and then switch to traditional ones

Example: GABIL system

- GABIL system[1] uses a GA to learn boolean concepts represented by a disjunctive set of propositional rules
- The learning tasks included artificial learning tasks designed to explore the systems' generalization accuracy and the real world problem of breast cancer diagnosis
- Algorithm uses $r = 0.6$ fraction of the parent population replaced by crossover. Mutation rate $m = 0.001$
- The population size p was varied from 100 to 1000, depending on the specific learning task
- Fitness function was $\texttt{Fitness}(h) = (correct(h))^2$

GABIL achieved average generalization accuracy of 92.1% over 12 synthetic problems; whereas the performance of the other systems (C4.5, ID5R, and AQ14) ranged from 91.2% to 96.6%

[1] Using genetic algorithms for concept learning, *De Jong, Kenneth A and Spears, William M and Gordon, Diana F*, In Genetic Algorithms for Machine Learning, pp 5–32, Springer(1993)

Thank You!

Thank you very much for your attention!

Queries ?

(Reference[2])

[2] Book - *Machine Learning*, ch-9, Tom Mitchell.