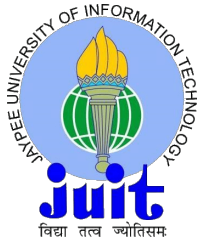

Machine Learning

(Course Code: 18B1WCI634)

Genetic Algorithms

Mr. Sandeep Kumar Patel

Assistant Professor

Jaypee University of Information Technology

Waknaghat, Solan, HP-173234

Genetic Algorithms

Definition

Genetic Algorithms (GAs) are search and optimization techniques inspired by the process of natural evolution.

- Hypotheses are represented as **bit strings**.
- A population of candidate solutions is evolved over time.
- Uses operations similar to biological evolution:
  - Selection
  - Crossover
  - Mutation

Motivation of Genetic Algorithms

- Evolution is a robust method for adaptation.
- Can handle complex hypothesis spaces.
- Useful when:
  - Relationships between variables are unknown
  - Search space is large and complex
- Easily parallelizable.

Key Concepts

- **Population (P)**: Set of candidate hypotheses
- **Chromosome**: Representation of hypothesis (bit string)
- **Fitness Function**: Evaluates quality of hypothesis
- **Generation**: Iteration of evolution

Genetic Algorithm Framework

Inputs

- Fitness function
- Population size (p)
- Crossover rate (r)
- Mutation rate (m)

Goal

Find hypothesis with maximum fitness

Genetic Algorithm Pseudocode

Algorithm

Initialize population P
Evaluate fitness
while max fitness ¡ threshold:

- Select individuals
- Apply crossover
- Apply mutation
- Update population

Return best hypothesis

Representation of Hypothesis

- Typically represented using **bit strings**
- Example:

$$10110 \rightarrow \text{encoded solution}$$

- Can represent:
  - Rules
  - Parameters
  - Features

Example: Rule Encoding

- Attribute: Outlook = {Sunny, Overcast, Rain}
- Encoding:
  - 100 $\rightarrow$ Sunny
  - 010 $\rightarrow$ Overcast
  - 001 $\rightarrow$ Rain
- Combined rule:

$$011 \rightarrow (Outlook = Overcast \vee Rain)$$

Fitness Function

Definition

A function that assigns a score to each hypothesis.

- Determines survival of individuals
- Example:
  - Classification accuracy
  - Number of games won

Selection in Genetic Algorithm

Definition

Selection is the process of choosing individuals from the population based on their fitness to produce the next generation.

- Inspired by natural selection
- Higher fitness $\rightarrow$ higher chance of selection
- Drives evolution toward better solutions

Why Selection is Important

- Improves overall population quality
- Preserves good solutions
- Eliminates weak candidates over time

Goal: Maximize fitness across generations

Fitness Proportionate Selection

Idea

Probability of selection is proportional to fitness.

Probability of selecting hypothesis h_i:

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

Interpretation:

- Higher fitness $\Rightarrow$ higher probability
- Better solutions are more likely to be selected

Roulette Wheel Selection

- Individuals placed on a wheel
- Each gets portion proportional to fitness
- Spin the wheel to select

Key Idea

Selection probability depends on relative fitness

Tournament Selection

Idea

Select k individuals randomly and choose the best among them.

- Repeat process to select parents
- k = tournament size

Advantages:

- Simple and efficient
- Works well in practice

Rank Selection

Idea

Individuals are ranked based on fitness, not actual values.

- Reduces dominance of very strong individuals
- Provides stable selection pressure

$$P(i) = \frac{\text{rank}(i)}{\sum \text{ranks}}$$

Comparison of Selection Methods

Method	Advantage	Limitation
Roulette Wheel	Simple	Sensitive to fitness scale
Tournament	Efficient	May reduce diversity
Rank Selection	Stable	Slightly slower

Crossover in Genetic Algorithm

Definition

Crossover is a genetic operator that combines two parent chromosomes to produce new offspring.

- Inspired by biological reproduction
- Helps explore new solutions
- Main types:
  - Single-point crossover
  - Two-point crossover
  - Uniform crossover

Single-Point Crossover

- One crossover point is selected
- Segments after the point are swapped

Example

Parent 1: 11111|00000

Parent 2: 00000|11111

Offspring 1: 11111 11111

Offspring 2: 00000 00000

Two-Point Crossover

- Two crossover points are selected
- Middle segment is exchanged

Example

Parent 1: 11|111|000

Parent 2: 00|000|111

Offspring 1: 11 000 000

Offspring 2: 00 111 111

Uniform Crossover

- Each gene is selected randomly from parents
- A mask controls selection

Example

Parent 1: 1 1 1 0 0

Parent 2: 0 0 0 1 1

Mask: 1 0 1 0 1

Offspring: 1 0 1 1 0

Comparison of Crossover Types

Type	Cut Points	Mixing Level
Single-point	1	Low
Two-point	2	Medium
Uniform	Multiple	High

Mutation in Genetic Algorithm

Definition

Mutation is a genetic operator that randomly alters one or more genes in a chromosome.

- Introduces randomness
- Maintains diversity in population
- Prevents premature convergence

Why Mutation is Important

- Avoids local optimum
- Ensures genetic diversity
- Helps explore new solutions

Goal: Maintain variation in population

Bit Flip Mutation

- Used for binary encoding
- Each bit may flip ($0 \leftrightarrow 1$)

Example

Before: 101100

After: 100100

Mutation Probability

Definition

Each gene is mutated with probability p_m.

$$P(\text{bit flips}) = p_m$$

Typical Values:

- Very small (0.001 to 0.01)

Interpretation:

- Small $p_m \rightarrow$ stable evolution
- Large $p_m \rightarrow$ random search behavior

Advantages and Limitations

Advantages:

- Maintains diversity
- Prevents stagnation

Limitations:

- Too much mutation $\rightarrow$ random search
- Too little mutation $\rightarrow$ premature convergence

Numerical Example: Genetic Algorithm

Problem: Maximize

$$f(x) = x^2, \quad x \in [0, 7]$$

Encoding:

- Use 3-bit binary representation

Initial Population

Binary	Decimal	Fitness $f(x)$
001	1	1
010	2	4
011	3	9
000	0	0

Selection Probability

$$P(i) = \frac{f_i}{\sum f_i}$$

Total Fitness $= 14$

- $001 \rightarrow 1/14$
- $010 \rightarrow 4/14$
- $011 \rightarrow 9/14$
- $000 \rightarrow 0$

Crossover

Parents:

011 010

After crossover:

010 011

Mutation

Before Mutation:

011

After Mutation:

111

$$x = 7, \quad f(x) = 49$$

Final Result

- Best solution found: $x = 7$
- Maximum fitness: $f(x) = 49$

Genetic Algorithm improves solutions over generations

Genetic Algorithms: Why Do They Work?

Key Idea

Genetic Algorithms work by identifying and combining useful patterns in candidate solutions.

- These patterns are called **Schemas**
- GA processes many schemas simultaneously
- Leads to efficient search in large spaces

Schema Representation

Definition

A **schema** is a template describing a subset of bit strings.

Symbols used:

- 0 or 1 $\rightarrow$ fixed value
- * $\rightarrow$ don't care

Example

Schema: `1*0*`

Represents: 1000, 1001, 1100, 1101

Interpretation:

- First bit must be 1
- Third bit must be 0
- Others can be anything

Schema Properties

- **Order (o(H))**: Number of fixed positions

$$H = 1 * 0 * 1 \Rightarrow o(H) = 3$$

- **Defining Length ($\delta(H)$)**: Distance between first and last fixed position

$$H = 1 * * * 1 \Rightarrow \delta(H) = 4$$

- **Fitness (f(H))**: Average fitness of strings matching schema

Selection Effect on Schema

Let:

- $m(H, t)$ = number of strings matching schema H at generation t
- $f(H)$ = average fitness of schema
- $\bar{f}$ = average population fitness

Expected Number After Selection

$$E[m(H, t+1)] = m(H, t) \cdot \frac{f(H)}{\bar{f}}$$

Explanation:

- If $f(H) > \bar{f} \rightarrow$ schema grows
- If $f(H) < \bar{f} \rightarrow$ schema decreases

Effect of Crossover

- Crossover may disrupt schema
- Probability depends on defining length

Survival Probability

$$P(\text{schema survives}) \geq 1 - \frac{\delta(H)}{l-1}$$

Where:

- $\delta(H)$ = defining length
- l = chromosome length

Interpretation:

- Short schemas $\rightarrow$ more stable
- Long schemas $\rightarrow$ likely to break

Effect of Mutation

- Mutation flips bits randomly
- Can destroy schema

Survival Probability

$$P(\text{schema survives}) \geq (1 - p_m)^{o(H)}$$

Where:

- p_m = mutation rate
- $o(H)$ = order of schema

Interpretation:

- Lower mutation $\rightarrow$ schema survives more
- Higher order $\rightarrow$ more chance of destruction

Schema Theorem

Theorem

$$E[m(H, t+1)] \geq m(H, t) \cdot \frac{f(H)}{\bar{f}} \cdot \left(1 - \frac{\delta(H)}{l-1}\right) \cdot (1 - p_m)^{o(H)}$$

Interpretation:

- Good schemas grow exponentially
- Short, low-order schemas survive better
- GA favors useful building blocks

Numerical Example: Schema Theorem

Given:

- $m(H, t) = 10$
- $f(H) = 8$
- $\bar{f} = 6$
- Chromosome length $l = 6$
- Defining length $\delta(H) = 2$
- Order $o(H) = 3$
- Mutation rate $p_m = 0.01$

Step 1: Selection Effect

$$E_1 = m(H, t) \cdot \frac{f(H)}{\bar{f}}$$

$$E_1 = 10 \cdot \frac{8}{6} = 13.33$$

Interpretation:

- Since $f(H) > \bar{f}$, schema increases

Step 2: Effect of Crossover

$$P_c = 1 - \frac{\delta(H)}{l-1}$$

$$P_c = 1 - \frac{2}{5} = 0.6$$

Interpretation:

- 60% chance schema survives crossover

Step 3: Effect of Mutation

$$P_m = (1 - p_m)^{o(H)}$$

$$P_m = (0.99)^3 \approx 0.9703$$

Interpretation:

- High survival probability due to low mutation rate

Step 4: Final Schema Theorem

$$E[m(H, t+1)] \geq E_1 \cdot P_c \cdot P_m$$

$$E[m(H, t+1)] \geq 13.33 \cdot 0.6 \cdot 0.9703$$

$$E[m(H, t+1)] \approx 7.75$$

Final Answer:

$$\boxed{E[m(H, t+1)] \approx 7.75}$$