Genetic Algorithms

Dr. Mahmoud Nabil Mahmoud
mnmahmoud@ncat.edu

North Carolina A & T State University

February 10, 2021

Agenda

GA a simulation by hand

For a function $f(x) = x^2$ where $x \in [0, 31]$ a population of 4 strings.

String No.	Initial Population (Randomly Generated)	x Value (Unsigned Integer)	$f(x)$ x^2	pselect, $\frac{f_i}{\Sigma f}$	Expected count $\frac{f_i}{\bar{f}}$	Actual Count from Roulette Wheel
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4.0
Average			293	0.25	1.00	1.0
Max			576	0.49	1.97	2.0

GA a simulation by hand

Reproduction, and Crossover with no mutation.

Mating Pool after Reproduction (Cross Site Shown)	Mate (Randomly Selected)	Crossover Site (Randomly Selected)	New Population	x Value	$f(x)$ x^2
0 1 1 0 | 1	2	4	0 1 1 0 0	12	144
1 1 0 0 | 0	1	4	1 1 0 0 1	25	625
1 1 | 0 0 0	4	2	1 1 0 1 1	27	729
1 0 | 0 1 1	3	2	1 0 0 0 0	16	256
					1754
					439
					729

- The population average fitness improved from 239 to 439.
- The maximum fitness also improved from 576 to 729

Outline

Similarity Schema

What information is contained in this population to guide directed search for improvement? (i.e., causal relation similarity/fitness)

String	Fitness
01101	169
11000	576
01000	64
10011	361

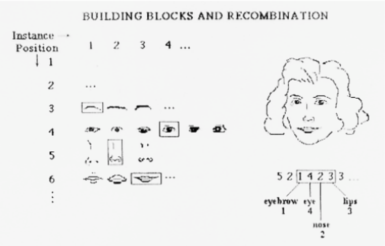
Similarity Schema

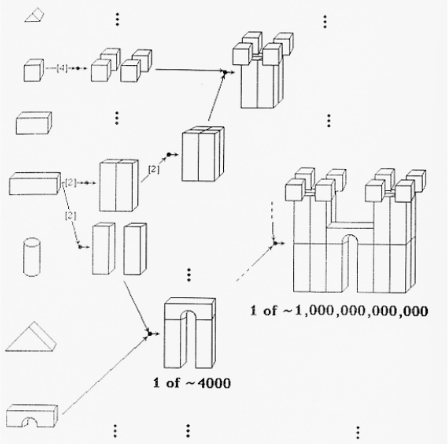
What information is contained in this population to guide directed search for improvement? (i.e., causal relation similarity/fitness)

String	Fitness
01101	169
11000	576
01000	64
10011	361

- Strings starts with 1 on the left seems to be better.
- Similar strings fall under the same schema (i.e., similarity template)
- Study of schemata proves the power of GA.

Similarity Schema





The Power of Building Blocks

Similarity templates (Schemeta)

Schema - Holland (1975)

Schema is a template that identifies a subset of strings with similarities at certain string positions.

E.g.

0*1100*

Note

We can think of it as a pattern matching device: a schema matches a particular string if at every location in the schema a 1 matches a 1 in the string, or a 0 matches a 0, or a * matches either.

E.g. For a binary alphabet {**0, 1**}, we motivate a schema by appending a special symbol *****, or **dont care symbol**, producing a ternary alphabet {**0, 1, ***} that allows us to build schemata.

Notation: String, Population

Consider strings to be constructed over the binary alphabet

$$V=\{0, 1\}$$

- **Strings** as capital letters
- **Individual characters** by lowercase letters subscripted by their position.

Example

A = 0111000 may be represented symbolically as:
$A = a_1 a_2 a_3 a_4 a_5 a_6 a_7$

Example

a_i represents a **gene** (binary feature or detector)
a_i **value** represents an **allele**
A(t)represents a population of strings at time (or generation) **t**

Notation: Schema

Consider a schema H taken from the three-letter alphabet:

$$V=\{ 0, 1, * \};$$

* **asterisk** is a **dont care symbol** which matches either a 0 or a 1 at a particular position.

Schema Matching

A **bit string** matches a particular **schemata** if that bit string can be constructed from the schemata by replacing the "*" symbol with the appropriate bit value.

Example

H = *11*0**

String A = 0111000

String A is an example of the schema H because the string alleles a_i match schema positions h_i at the fixed positions 2, 3 and 5.

Outline

Schema Properties

Defining Length of Schema

$\delta(H)$ is the distance between the first and last fixed string position

Ex.

- $H = 011 * 1 * *$
- $\delta(H) = 5 - 1 = 4$

Order of Schema:

$o(H)$ is the number of fixed positions present in the template

- $H = 0 * * * * * *$
- $\delta(H) = 0$ because there is only one fixed position
- $o(H) = 1$

Schema Properties

Note

Schemata and their properties serve as notational devices for rigorously discussing and classifying string similarities.

Note

They provide the basic means for analyzing the **net effect of reproduction and genetic operators** on the building blocks contained within the population.

Schema Counting

How many possible schemata for binary string population of length ($\ell = 5$)

Schema Counting

How many possible schemata for binary string population of length ($\ell = 5$)

- Each character can be from {0, 1, *}

Schema Counting

How many possible schemata for binary string population of length ($\ell = 5$)

- Each character can be from $\{0, 1, *\}$
- Thus, we have $3 \times 3 \times 3 \times 3 \times 3 = 3^5 = 243$ possible schemata

Schema Counting

How many possible schemata for binary string population of length ($\ell = 5$)

- Each character can be from $\{0, 1, *\}$
- Thus, we have $3 \times 3 \times 3 \times 3 \times 3 = 3^5 = 243$ possible schemata
- Alphabet of cardinality k has $(k + 1)^\ell$ possible schemata

Schema Counting

Consider the binary string 10, how many schemata cover this string?

Schema Counting

Consider the binary string 10, how many schemata cover this string?

$$\{**,*0,1*,10\}$$

- For string **s** of length ℓ, 2^ℓ schemata cover **s**.

Schema Counting

Consider the binary string 10, how many schemata cover this string?

$$\{**,*0,1*,10\}$$

- For string **s** of length ℓ, 2^ℓ schemata cover **s**.
- A population of size **n** may contains somewhere between 2^ℓ to $n \times 2^\ell$ possible schemata (over estimate).

Schema Counting

Consider the binary string 10, how many schemata cover this string?

$$\{**,*0,1*,10\}$$

- For string **s** of length ℓ, 2^ℓ schemata cover **s**.
- A population of size **n** may contains somewhere between 2^ℓ to $n \times 2^\ell$ possible schemata (over estimate).
- Not all this large numbers of schemata are explored due to the destructive effect of cross over and mutation.

Schema Counting

Consider the binary string 10, how many schemata cover this string?

$$\{**,*0,1*,10\}$$

- For string **s** of length ℓ, 2^ℓ schemata cover **s**.
- A population of size **n** may contains somewhere between 2^ℓ to $n \times 2^\ell$ possible schemata (over estimate).
- Not all this large numbers of schemata are explored due to the destructive effect of cross over and mutation.
- What we will proof is that a population of size n explore at least $O(n^3)$ schemeta.

Schema Counting

Consider the binary string 10, how many schemata cover this string?

$$\{**,*0,1*,10\}$$

- For string **s** of length ℓ, 2^ℓ schemata cover **s**.
- A population of size **n** may contains somewhere between 2^ℓ to $n \times 2^\ell$ possible schemata (over estimate).
- Not all this large numbers of schemata are explored due to the destructive effect of cross over and mutation.
- What we will proof is that a population of size n explore at least $O(n^3)$ schemeta.

Schema Counting

Consider the binary string 10, how many schemata cover this string?

$$\{**,*0,1*,10\}$$

- For string **s** of length ℓ, 2^ℓ schemata cover **s**.
- A population of size **n** may contains somewhere between 2^ℓ to $n \times 2^\ell$ possible schemata (over estimate).
- Not all this large numbers of schemata are explored due to the destructive effect of cross over and mutation.
- What we will proof is that a population of size n explore at least $O(n^3)$ schemeta.

How many schemata are usefully processed?
(is there a lower bound linked to the population size n)?
Holland $o(n^3)$.

Outline

Effect of Reproduction on Schemata

Suppose at time **t**, there are **m** examples of a particular schema **H** in population **A(t)**

$$m = m(H, t)$$

Effect of Reproduction on Schemata

Suppose at time **t**, there are **m** examples of a particular schema **H** in population **A(t)**

$$m = m(H, t)$$

During reproduction, a string A_i gets copied according to its fitness with probability $p_i = \frac{f_i}{\sum f_i}$

$$m(H, t+1) = m(H, t) \times n \times \frac{f(H)}{\sum f_i}$$

f(H) is the **average fitness** of the strings representing schema **H** at time **t**.

Effect of Reproduction on Schemata

we may write the reproductive schema growth equation as:

$$m(H, t+1) = m(H, t) \times n \times \frac{f(H)}{\sum f_i}$$

Simplification

- If we recognize that the average fitness of the entire population as $\overline{f} = \frac{\sum f_i}{n}$
- we may express the reproductive schema growth equation as:

$$m(H, t+1) = m(H, t) \times \frac{f(H)}{\overline{f}}$$

Effect of Reproduction on Schemata

Reproductive schema growth equation:

$$m(H, t+1) = m(H, t) \times \frac{f(H)}{\bar{f}}$$

- A particular schema grows as the ratio of the average fitness of the schema to the average fitness of the population

Effect of Reproduction on Schemata

Reproductive schema growth equation:

$$m(H, t+1) = m(H, t) \times \frac{f(H)}{\bar{f}}$$

- A particular schema grows as the ratio of the average fitness of the schema to the average fitness of the population
- Schemata with **fitness values** above the population average will receive an increasing number of samples in the next generation.

Effect of Reproduction on Schemata

Reproductive schema growth equation:

$$m(H, t+1) = m(H, t) \times \frac{f(H)}{\bar{f}}$$

- A particular schema grows as the ratio of the average fitness of the schema to the average fitness of the population
- Schemata with **fitness values** above the population average will receive an increasing number of samples in the next generation.
- Schemata with **fitness values** below the population average will receive a decreasing number of samples.

Effect of Reproduction on Schemata

Reproductive schema growth equation:

$$m(H, t+1) = m(H, t) \times \frac{f(H)}{\bar{f}}$$

- A particular schema grows as the ratio of the average fitness of the schema to the average fitness of the population
- Schemata with **fitness values** above the population average will receive an increasing number of samples in the next generation.
- Schemata with **fitness values** below the population average will receive a decreasing number of samples.
- **All the schemata** in a population grow or decay according to their schema averages under the operation of reproduction alone.

Reproductive schema growth equation:

$$m(H, t+1) = m(H, t) \times \frac{f(H)}{\bar{f}}$$

Reproductive schema growth equation:

$$m(H, t+1) = m(H, t) \times \frac{f(H)}{\overline{f}}$$

Suppose we assume that a particular schema **H** remains above average an amount $c.\overline{f}$ with a **c** constant. Under this assumption, we can write:

Reproductive schema growth equation:

$$m(H, t+1) = m(H, t) \times \frac{f(H)}{\overline{f}}$$

Suppose we assume that a particular schema **H** remains above average an amount $c.\overline{f}$ with a **c** constant. Under this assumption, we can write:

$$m(H, t+1) = m(H, t) \times \frac{\overline{f}+c\overline{f}}{\overline{f}} = m(H, t) \times (1+c)$$

Starting at **t=0**, and assuming a stationary value of **c**, we obtain the equation:

$$m(H, t+1) = m(H, 0) \times (1+c)^t$$

Note

Reproduction allocates exponentially increasing (decreasing) numbers of trials to above (below) average schema.

Quantitave Effect of Reproduction on Schemata

$$m(H, t+1) = m(H, 0) \times (1 + c)^t$$

- Reproduction can allocate exponentially increasing and decreasing numbers of schemata to future generations in parallel.

Quantitave Effect of Reproduction on Schemata

$$m(H, t+1) = m(H, 0) \times (1+c)^t$$

- Reproduction can allocate exponentially increasing and decreasing numbers of schemata to future generations in parallel.
- Many different schemata are sampled in **parallel** according to the same rule through the use of **n** simple reproduction operations.

Quantitave Effect of Reproduction on Schemata

$$m(H, t+1) = m(H, 0) \times (1+c)^t$$

- Reproduction can allocate exponentially increasing and decreasing numbers of schemata to future generations in parallel.
- Many different schemata are sampled in **parallel** according to the same rule through the use of **n** simple reproduction operations.
- However, **reproduction does not promote exploration of new regions of the search space**.

Quantitave Effect of Reproduction on Schemata

$$m(H, t+1) = m(H, 0) \times (1 + c)^t$$

- Reproduction can allocate exponentially increasing and decreasing numbers of schemata to future generations in parallel.
- Many different schemata are sampled in **parallel** according to the same rule through the use of **n** simple reproduction operations.
- However, **reproduction does not promote exploration of new regions of the search space**.
- This is where **crossover** steps in

Effect of Crossover on Schemata

Consider a particular string of length $\ell = 7$ and two representative schemata within that string:

$$A = 0111000$$
$$H_1 = *1****0$$
$$H_2 = ***10**$$

Effect of Crossover on Schemata

Consider a particular string of length $\ell = 7$ and two representative schemata within that string:

$$A = 0111000$$
$$H_1 = *1****0$$
$$H_2 = ***10**$$

Recall: Crossover Operation

- crossover proceeds with the random selection of a mate;
- Random selection of a crossover site
- The exchange of substrings from the beginning of the string to the crossover site inclusively with the corresponding substring of the chosen mate.

Effect of Crossover on Schemata

Assuming that we have the following randomly chosen **crossover site: 3**

$$A = 011|1000$$
$$H_1 = *1*|***0$$
$$H_2 = ***|10**$$

- H_1 is **destroyed**. Defining length $= 5$
- H_2 will **survive**. Defining length $= 1$

Note

H_1 is less likely to survive crossover than schema H_2 because on average the crossover site is more likely to fall between the extreme fixed positions.

Lower Bound on Crossover Survival Probability

$$A = 011|1000$$
$$H_1 = *1*|***0$$
$$H_2 = ***|10**$$

Lower Bound on Crossover Survival Probability

$$A = 011|1000$$
$$H_1 = *1*|***0$$
$$H_2 = ***|10**$$

- $p_d(H_1) = \frac{\delta(H_1)}{\ell - 1} = \frac{5}{6}$
- $p_d(H_2) = \frac{\delta(H_2)}{\ell - 1} = \frac{1}{6}$
- $p_s(H_1) = 1 - p_d(H_1) = \frac{1}{6}$
- $p_s(H_2) = 1 - p_d(H_2) = \frac{5}{6}$

To generalize, a schema survives when the cross over site falls outside the defining length. The survival probability under simple crossover is $p_s(H)$

$$p_s(H) = 1 - \frac{\delta(H)}{\ell - 1}$$

Lower Bound on Crossover Survival Probability

If we consider the probability of performing a crossover operation to be p_c

$$p_s(H) = 1 - p_c\left(\frac{\delta(H)}{\ell-1}\right)$$

- Independence is assumed between the two event (crossover and schemata destruction)

Combined Effect of Reproduction and Crossover

Assuming independence of the **reproduction** and **crossover** operations.

$$m(H, t+1) = m(H, 0) \times (1+c)^t \times \left[1 - p_c\left(\frac{\delta(H)}{\ell - 1}\right)\right]$$

Schema **H** grows or decays depending upon a multiplication factor.

Combined Effect of Reproduction and Crossover

Assuming independence of the **reproduction** and **crossover** operations.

$$m(H, t+1) = m(H, 0) \times (1+c)^t \times \left[1 - p_c\left(\frac{\delta(H)}{\ell - 1}\right)\right]$$

Schema **H** grows or decays depending upon a multiplication factor.

That factor depends on 2 things:

- whether the schema is above or below the population average
- whether the schema has relatively short or long defining length

Combined Effect of Reproduction and Crossover

Assuming independence of the **reproduction** and **crossover** operations.

$$m(H, t+1) = m(H, 0) \times (1+c)^t \times \left[1 - p_c\left(\frac{\delta(H)}{\ell - 1}\right)\right]$$

Schema **H** grows or decays depending upon a multiplication factor.

That factor depends on 2 things:

- whether the schema is above or below the population average
- whether the schema has relatively short or long defining length

Note

Clearly, those schemata with both above-average observed performance and short defining lengths are going to be sampled at exponentially increasing rates.

Effect of Mutation

Mutation is the random alteration of a single position with probability p_m

Effect of Mutation

Mutation is the random alteration of a single position with probability p_m

In order for a schema **H** to survive, all of the specified positions must themselves survive.

Effect of Mutation

Mutation is the random alteration of a single position with probability p_m

In order for a schema **H** to survive, all of the specified positions must themselves survive.

A single allele survives with a probability $(1 - p_m)$

Effect of Mutation

Mutation is the random alteration of a single position with probability p_m

In order for a schema **H** to survive, all of the specified positions must themselves survive.

A single allele survives with a probability $(1 - p_m)$

Since each of the mutations is **statistically independent**, a particular schema **H** survives when each of the **o(H)** fixed positions within the schema survives

Effect of Mutation

Mutation is the random alteration of a single position with probability p_m

In order for a schema **H** to survive, all of the specified positions must themselves survive.

A single allele survives with a probability $(1 - p_m)$

Since each of the mutations is **statistically independent**, a particular schema **H** survives when each of the **o(H)** fixed positions within the schema survives

The survival probability is multiplied by itself o(H) times:

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

Effect of Mutation

Mutation is the random alteration of a single position with probability p_m

In order for a schema **H** to survive, all of the specified positions must themselves survive.

A single allele survives with a probability $(1 - p_m)$

Since each of the mutations is **statistically independent**, a particular schema **H** survives when each of the **o(H)** fixed positions within the schema survives

The survival probability is multiplied by itself o(H) times:

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

For small values of p_m ($p_m << 1$), we can write:

$$(1 - o(H)p_m)$$

Fundamental Theorem of Genetic Algorithms

$$m(H, t+1) \geq m(H, t) \times \frac{f(H)}{\overline{f}} \times \left[1 - p_c\left(\frac{\delta(H)}{\ell - 1}\right) - o(H)p_m\right]$$

- $m(H, t+1)$ Expected Count of Schema H at time (t+1)
- $m(H, t)$ Expected Count of Schema H at time (t)
- $\frac{f(H)}{\overline{f}}$ ratio of schema fitness to the total fitness
- $\left[1 - p_c\left(\frac{\delta(H)}{\ell - 1}\right) - o(H)p_m\right]$ Survival probability

Who shall live and who shall die?

Short, low-order, above-average schemata are given exponentially increasing trials in subsequent generations (building blocks)

Outline

Schema processing by hand

Let us observe how the GA processes schemata not individual strings-within the population

Let us consider three particular schemata, H_1, H_2 and H_3 Where

- $H_1 = 1****$
- $H_2 = *10**$
- $H_3 = 1***0$

Observe the effect of reproduction, crossover, and mutation.

Hand Calculations

String Processing

String No.	Initial Population (Randomly Generated)	x Value (Unsigned Integer)	$f(x)$ x^2	pselect, $\frac{f_i}{\Sigma f}$	Expected count $\frac{f_i}{\bar{f}}$	Actual Count from Roulette Wheel
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4.0
Average			293	0.25	1.00	1.0
Max			576	0.49	1.97	2.0

Schema Processing

		Before Reproduction	
		String Representatives	Schema Average Fitness $f(H)$
H_1	1 * * * *	2,4	469
H_2	* 1 0 * *	2,3	320
H_3	1 * * * 0	2	576

Hand Calculations

String Processing

Mating Pool after Reproduction (Cross Site Shown)	Mate (Randomly Selected)	Crossover Site (Randomly Selected)	New Population	x Value	$f(x)$ x^2
0 1 1 0 | 1	2	4	0 1 1 0 0	12	144
1 1 0 0 | 0	1	4	1 1 0 0 1	25	625
1 1 | 0 0 0	4	2	1 1 0 1 1	27	729
1 0 | 0 1 1	3	2	1 0 0 0 0	16	256
Sum					1754
Average					439
Max					729

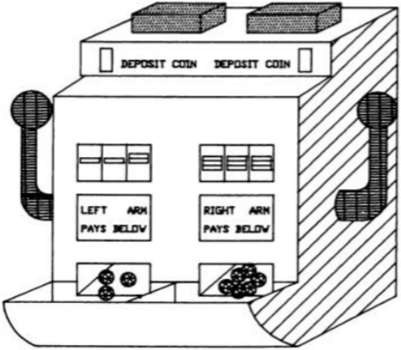
Schema Processing

After Reproduction			After All Operators		
Expected Count	Actual Count	String Representatives	Expected Count	Actual Count	String Representatives
3.20	3	2,3,4	3.20	3	2,3,4
2.18	2	2,3	1.64	2	2,3
1.97	2	2,3	0.0	1	4

Outline

Two Armed Bandit Problem

- Suppose a two armed slot machine where one arm pays a reward μ_1 and variance σ_1 and the other arm pays μ_2 and variance σ_2.
- where $\mu_1 \geq \mu_2$ Which arm should we play?



Two Armed Bandit Problem

- We can give each arm a try or some trials then play with the arm that pay more.

Two Armed Bandit Problem

- We can give each arm a try or some trials then play with the arm that pay more.
- This is known as a trade-off between the exploration for knowledge and the exploitation of that knowledge.

Two Armed Bandit Problem

- We can give each arm a try or some trials then play with the arm that pay more.
- This is known as a trade-off between the exploration for knowledge and the exploitation of that knowledge.
- Suppose we have a total of N trials to allocate among the two arms. We first allocate an equal number of trials n ($2n \leq N$) trials to each of the two arms.

Two Armed Bandit Problem

we can calculate the expected loss:

$$L(N, n) = |\mu_1 - \mu_2|.\left[(N - n)q + n(1 - q)\right]$$
$$\text{Where } q \approx \frac{1}{\sqrt{2.\pi}} \frac{e^{-x^2/2}}{x} \text{ and } x = \frac{\mu_1 - \mu_2}{\sqrt{\sigma_1^2 + \sigma_2^2}}.\sqrt{n}$$

Two Armed Bandit Problem

we can calculate the expected loss:

$$L(N, n) = |\mu_1 - \mu_2|.\left[(N - n)q + n(1 - q)\right]$$
$$\text{Where } q \approx \frac{1}{\sqrt{2.\pi}}\frac{e^{-x^2/2}}{x} \text{ and } x = \frac{\mu_1 - \mu_2}{\sqrt{\sigma_1^2 + \sigma_2^2}}.\sqrt{n}$$

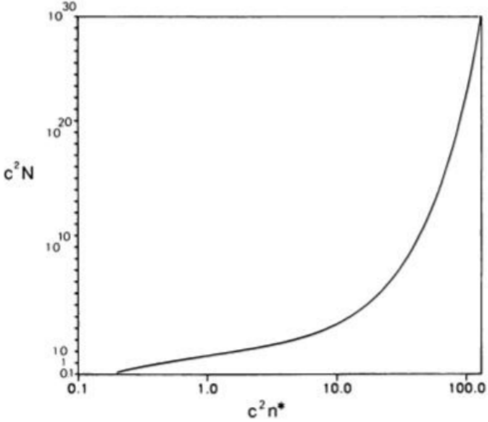
two sources of loss are associated with the procedure.

- The first loss is a result of issuing n trials to the wrong arm during the experiment.
- The second is a result of choosing the arm associated with the lower payoff even after performing the experiment.

if N, μ_1, μ_2, σ_1, σ_2 are known, how to get optimal n^*

Holland (1975) has performed calculations that show how trials should be allocated between the two arms to minimize expected losses.

- $n^* \approx b^2 ln\left[\frac{N^2}{8\pi b^4 lnN^2}\right]$
- $N \approx \sqrt{8\pi b^4 lnN^2} e^{n*/2b^2}$



We should give slightly more than exponentially increasing trials to the observed best arm. The same conclusion apply to the k-armed bandit.

GA and K-Armed Bandit Problem

- In the usual GA we consider the simultaneous solution of many multi-armed bandits.

GA and K-Armed Bandit Problem

- In the usual GA we consider the simultaneous solution of many multi-armed bandits.
- if we consider a set of competing schemata as a particular k-armed bandit.

GA and K-Armed Bandit Problem

- In the usual GA we consider the simultaneous solution of many multi-armed bandits.
- if we consider a set of competing schemata as a particular k-armed bandit.
- Two schemata A and B are competing if they have the same * positions and the same fixed positions.

GA and K-Armed Bandit Problem

- In the usual GA we consider the simultaneous solution of many multi-armed bandits.
- if we consider a set of competing schemata as a particular k-armed bandit.
- Two schemata A and B are competing if they have the same * positions and the same fixed positions.

GA and K-Armed Bandit Problem

- In the usual GA we consider the simultaneous solution of many multi-armed bandits.
- if we consider a set of competing schemata as a particular k-armed bandit.
- Two schemata A and B are competing if they have the same * positions and the same fixed positions.

E.x.

There are eight $= 2^3$ competing schemata over the three positions 2, 3, and 5

```
* 0 0 * 0 * *
* 0 0 * 1 * *
* 0 1 * 0 * *
* 0 1 * 1 * *
* 1 0 * 0 * *
* 1 0 * 1 * *
* 1 1 * 0 * *
* 1 1 * 1 * *
```

GA and K-Armed Bandit Problem

Competing schemata

```
* 0 0 * 0 * *
* 0 0 * 1 * *
* 0 1 * 0 * *
* 0 1 * 1 * *
* 1 0 * 0 * *
* 1 0 * 1 * *
* 1 1 * 0 * *
* 1 1 * 1 * *
```

- Since these schemata are defined over the same positions, they compete with one another for precious population slots.

GA and K-Armed Bandit Problem

Competing schemata

```
* 0 0 * 0 * *
* 0 0 * 1 * *
* 0 1 * 0 * *
* 0 1 * 1 * *
* 1 0 * 0 * *
* 1 0 * 1 * *
* 1 1 * 0 * *
* 1 1 * 1 * *
```

- Since these schemata are defined over the same positions, they compete with one another for precious population slots.
- Each schema will grow or decay exponentially based on its average fitness.

GA and K-Armed Bandit Problem

Competing schemata

*	0	0	*	0	*	*
*	0	0	*	1	*	*
*	0	1	*	0	*	*
*	0	1	*	1	*	*
*	1	0	*	0	*	*
*	1	0	*	1	*	*
*	1	1	*	0	*	*
*	1	1	*	1	*	*

- Since these schemata are defined over the same positions, they compete with one another for precious population slots.
- Each schema will grow or decay exponentially based on its average fitness.
- The difference here we assumed only 8 schemata over three positions (i.e., 8-arm bandit)

GA and K-Armed Bandit Problem

Competing schemata

```
* 0 0 * 0 * *
* 0 0 * 1 * *
* 0 1 * 0 * *
* 0 1 * 1 * *
* 1 0 * 0 * *
* 1 0 * 1 * *
* 1 1 * 0 * *
* 1 1 * 1 * *
```

- Since these schemata are defined over the same positions, they compete with one another for precious population slots.
- Each schema will grow or decay exponentially based on its average fitness.
- The difference here we assumed only 8 schemata over three positions (i.e., 8-arm bandit)
- In GA we have a number of problems proceeding in parallel

GA and K-Armed Bandit Problem

- with three positions fixed over a string of length 7 there are

GA and K-Armed Bandit Problem

- with three positions fixed over a string of length 7 there are
  - C(7,3)= 35, 8-arm bandit

GA and K-Armed Bandit Problem

- with three positions fixed over a string of length 7 there are
  - C(7,3)= 35, 8-arm bandit
- with j positions fixed over a string of length ℓ there are

  C(ℓ,j), 2^j-arm bandit

GA and K-Armed Bandit Problem

- with three positions fixed over a string of length 7 there are
  - C(7,3)= 35, 8-arm bandit
- with j positions fixed over a string of length ℓ there are

  C(ℓ,j), 2^j-arm bandit
- Total number of problems processed in parallel a.k.a implicit parallelism

$$\sum_j C(\ell, j) = 2^\ell$$

GA and K-Armed Bandit Problem

- with three positions fixed over a string of length 7 there are
  - C(7,3)= 35, 8-arm bandit
- with j positions fixed over a string of length ℓ there are

  C(ℓ,j), 2^j-arm bandit
- Total number of problems processed in parallel a.k.a implicit parallelism

$$\sum_j C(\ell, j) = 2^\ell$$

- Not all problems are played equally due to the destructive effect of crossover and mutation

Outline

How many schema are processed usefully?

- Consider a population of n binary strings of length ℓ.

How many schema are processed usefully?

- Consider a population of n binary strings of length ℓ.
- Probability of given schema survival should be greater than certain threshold ϵ

How many schema are processed usefully?

- Consider a population of n binary strings of length ℓ.
- Probability of given schema survival should be greater than certain threshold ϵ
- As a result, assuming the operation of simple crossover and small mutation rate we consider only those schemata with error rate less than ϵ

$$p_c\left(\frac{\delta(H)}{\ell-1} - O(H)p_m\right) \leq \epsilon$$
$$\delta(H) \lessapprox \epsilon(\ell-1)$$

How many schema are processed usefully?

- Consider a population of n binary strings of length ℓ.
- Probability of given schema survival should be greater than certain threshold ϵ
- As a result, assuming the operation of simple crossover and small mutation rate we consider only those schemata with error rate less than ϵ

$$p_c\left(\frac{\delta(H)}{\ell-1} - O(H)p_m\right) \leq \epsilon$$
$$\delta(H) \lessapprox \epsilon(\ell - 1)$$

- Holland suggested with a particular defining length, we can estimate a lower bound on the number of unique schemata processed by an initially random population of strings.

How many schema are processed usefully? $O(n^3)$ proof

- Lets start by counting the number of schemata of defining length $\delta(H)$ that cover a single string in the population.

e.g. $\ell = 10$ and $\delta(H) = 4$

1 0 1 1 1 0 0 0 1 0

How many schema are processed usefully? $O(n^3)$ proof

- Lets start by counting the number of schemata of defining length $\delta(H)$ that cover a single string in the population.

e.g. $\ell = 10$ and $\delta(H) = 4$

1 0 1 1 1 0 0 0 1 0

- Schemeta count in a single window of size $\delta(H)$

$$2^{\delta(H)-1}$$

How many schema are processed usefully? $O(n^3)$ proof

- Lets start by counting the number of schemata of defining length $\delta(H)$ that cover a single string in the population.

e.g. $\ell = 10$ and $\delta(H) = 4$

1 0 1 1 1 0 0 0 1 0

- Schemeta count in a single window of size $\delta(H)$

$$2^{\delta(H)-1}$$

- Schemeta count for a string of length ℓ

$$2^{\delta(H)-1}.(\ell - \delta(H) + 1)$$

How many schema are processed usefully? $O(n^3)$ proof

- Lets start by counting the number of schemata of defining length $\delta(H)$ that cover a single string in the population.

e.g. $\ell = 10$ and $\delta(H) = 4$

1 0 1 1 1 0 0 0 1 0

- Schemeta count in a single window of size $\delta(H)$

$$2^{\delta(H)-1}$$

- Schemeta count for a string of length ℓ

$$2^{\delta(H)-1}.(\ell - \delta(H) + 1)$$

- For the entire population of size n we have

$$n_s = n.2^{\delta(H)-1}.(\ell - \delta(H) + 1)$$

How many schema are processed usefully? $O(n^3)$ proof

- Lets start by counting the number of schemata of defining length $\delta(H)$ that cover a single string in the population.

e.g. $\ell = 10$ and $\delta(H) = 4$

1 0 1 1 1 0 0 0 1 0

- Schemeta count in a single window of size $\delta(H)$
$$2^{\delta(H)-1}$$
- Schemeta count for a string of length ℓ
$$2^{\delta(H)-1}.(\ell - \delta(H) + 1)$$
- For the entire population of size n we have
$$n_s = n.2^{\delta(H)-1}.(\ell - \delta(H) + 1)$$
- Holland assumed initial population size $n = 2^{\delta(H)/2}$ or $n^2 = 2^{\delta(H)}$, then
$$n_s = n^3.\frac{\ell - \delta(H) + 1}{2} = O(n^3)$$

Criticize on Holland n^3 argument

- The estimate for n_s depends upon a particular choice of population size and any deviation in population size invalidates the derivation.
- Increase in population size decreases the exponent, thereby decreasing the apparent leverage.

Better Approximate

- Probabilistic estimate developed by Goldberg in 1985

Better Approximate

- Probabilistic estimate developed by Goldberg in 1985
- Consider a schema of order j in a population of size n, where $\left(p_{\text{one}} = p_{\text{zero}} = \frac{1}{2}\right)$

$$p\left(_{\texttt{single match on order j schema}}\right) = \left(\frac{1}{2}\right)^{j}$$

Better Approximate

- Probabilistic estimate developed by Goldberg in 1985
- Consider a schema of order j in a population of size n, where $\left(p_{\text{one}} = p_{\text{zero}} = \frac{1}{2}\right)$

$$p\left(\texttt{single match on order j schema}\right) = \left(\frac{1}{2}\right)^{j}$$

- The chance of having all population failure on this order j schema

$$p\left(\texttt{all failed matches of order j schema on population os size m}\right) = \left[1 - \left(\frac{1}{2}\right)^{j}\right]^{m}$$

Better Approximate

- Probabilistic estimate developed by Goldberg in 1985
- Consider a schema of order j in a population of size n, where $\left(p_{\text{one}} = p_{\text{zero}} = \frac{1}{2}\right)$

$$p\left(\texttt{single match on order j schema}\right) = \left(\frac{1}{2}\right)^{j}$$

- The chance of having all population failure on this order j schema

$$p\left(\texttt{all failed matches of order j schema on population os size m}\right) = \left[1 - \left(\frac{1}{2}\right)^{j}\right]^{m}$$

- The probability of one or more successes is then given by

$$p\left(\texttt{at least one order j success in size m population}\right) = 1 - \left[1 - \left(\frac{1}{2}\right)^{j}\right]^{m}$$

New Estimate

- There are $C(\ell, j)$ different ways to select the j positions in a string of length ℓ. Moreover, j fixed positions there are 2^j different schemata (a 0 or 1 at any of the j positions) thus expect to have the following number of schemata with one or more representatives in a population of size m:

$$\binom{\ell}{j} 2^j \left[1 - \left[1 - \left(\frac{1}{2}\right)^j\right]^m\right]$$

New Estimate

- There are $C(\ell, j)$ different ways to select the j positions in a string of length ℓ. Moreover, j fixed positions there are 2^j different schemata (a 0 or 1 at any of the j positions) thus expect to have the following number of schemata with one or more representatives in a population of size m:

$$\binom{\ell}{j}2^j\left[1-\left[1-\left(\tfrac{1}{2}\right)^j\right]^m\right]$$

- To get the total number of such schemata we simply sum over the order from 1 to the string length ℓ:

$$\sum_{j=1}^{j=\ell}\binom{\ell}{j}2^j\left[1-\left[1-\left(\tfrac{1}{2}\right)^j\right]^m\right]$$

New Estimate

- There are $C(\ell, j)$ different ways to select the j positions in a string of length ℓ. Moreover, j fixed positions there are 2^j different schemata (a 0 or 1 at any of the j positions) thus expect to have the following number of schemata with one or more representatives in a population of size m:

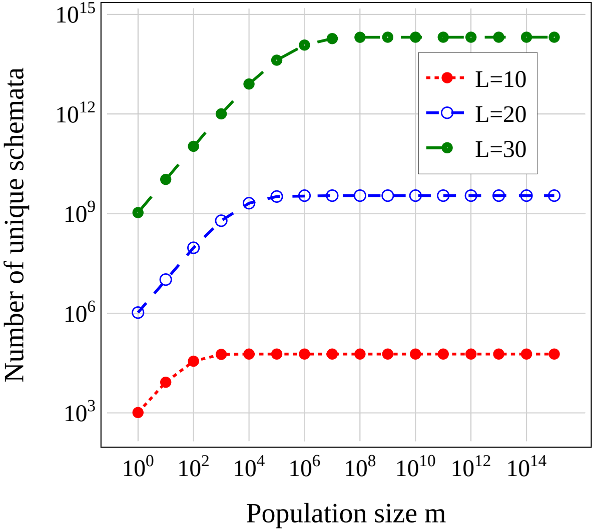
$$\binom{\ell}{j} 2^j \left[1 - \left[1 - \left(\tfrac{1}{2}\right)^j\right]^m\right]$$

- To get the total number of such schemata we simply sum over the order from 1 to the string length ℓ:

$$\sum_{j=1}^{j=\ell} \binom{\ell}{j} 2^j \left[1 - \left[1 - \left(\tfrac{1}{2}\right)^j\right]^m\right]$$

- No closed form formula or asyeptotic relation has been discovered for this expression

New Estimate



Outline

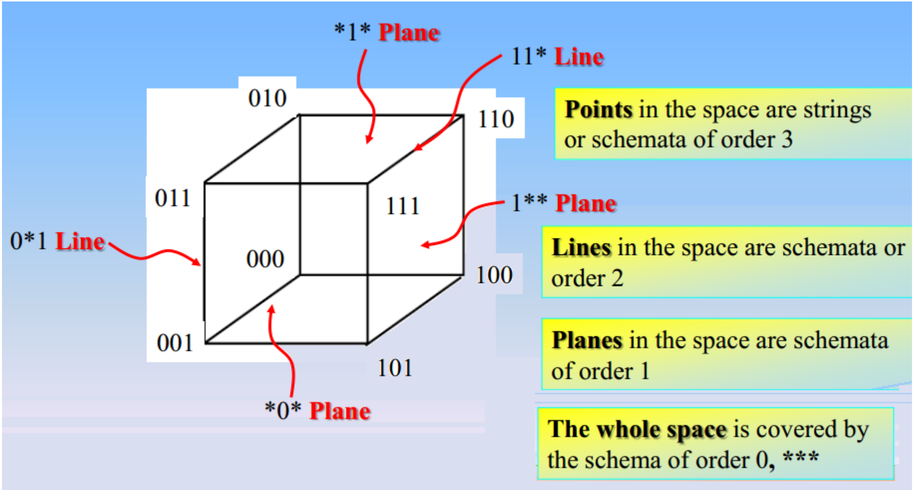
Hyperplane

Hyperplane

A hyperplane is a concept in geometry. It is a generalization of the concept of a plane.

- In 1-D space (such as a line), a hyperplane is a point; it divides a line into two rays.
- In 2-D space (such as the xy plane), a hyperplane is a line; it divides the plane into two half-planes.
- In 3-D space a hyperplane is an ordinary plane; it divides the space into two half-spaces.
- This concept can also be applied to four-dimensional space and beyond, where the dividing object is simply referred to as a hyperplane

Visualization of Schemata as Hyperplanes in 3-D Space

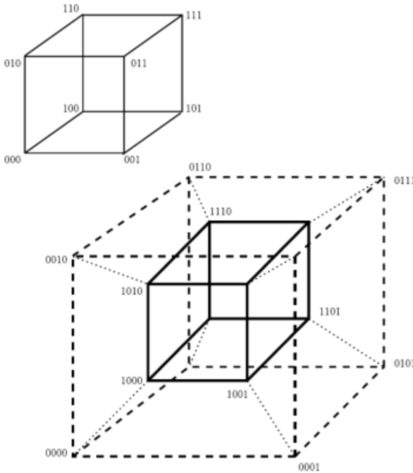


We can think of a GA cutting across different hyperplanes to search for improved performance.

Schemeta as Hyperplane

All adjacent corners are labeled by bit strings that differ by exactly 1bit
This creates an assignment to the points in hyperspace that gives the proper adjacency in the space between strings that are 1 bit different.

- inner cube: corresponds to 1***
- outer cube corresponds to 0***
- fronts of both cubes: *0**
- front of the inner cube: order-2 hyperplane 10**



References

- Jakub Konecny, Federated Learning Privacy-Preserving Collaborative Machine Learning without Centralized Training Data
- H. B. McMahan, et al. Communication-Efficient Learning of Deep Networks from Decentralized Data. AISTATS 2017
- K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, K. Seth. Practical Secure Aggregation for Privacy-Preserving Machine Learning, CCS 2017.

Questions