UNIT 5

Machine Learning

Contents

Concept of Learning

Learning

- Acquiring new knowledge – Knowledge acquisition
- Modifying Old Knowledge, behavior, and skills- skills refinement
- May involve synthesizing different types of information
- Learning is also based on feedbacks

Concept of Learning

Types of Learning

- Rote Learning
  - Rote learning is a technique which focuses on memorization
  - It avoids understanding the inner complexities and inferences of the subject that is being learned and instead focuses on memorizing the material
  - The major practice involved in rote learning is repetition
- Learning by examples
  - Agent learns by seeing examples and classify the similar object to same class
- Explanation based Learning
  - Describes objects in brief
- Learning by taking advice
- Learning by analogy

Concept of Learning

Types of Learning

- Inductive Learning
  - Determine general pattern, rules and facts
  - Mainly example of experienced based learning
- Deductive Learning
  - Determine specific patterns, rules and facts
  - New rules are generated from old ones

Concept of Learning

Machine Learning

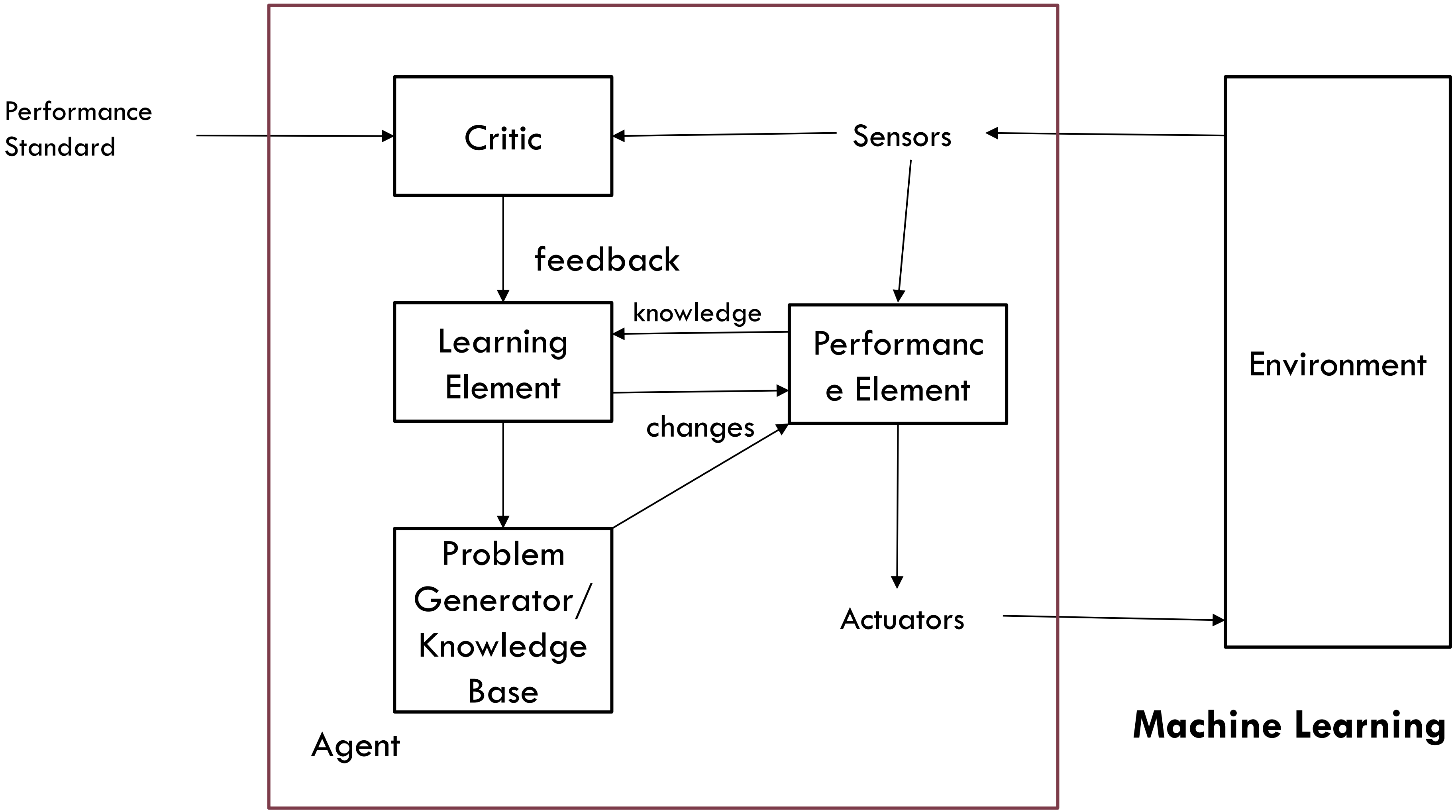
- Branch of AI that use algorithms to allow computer to evolve behaviours based on data collected from database or gathered through sensors
- Focuses on prediction, based on known properties learned from the training data
- The performance is usually evaluated with respect to the ability to reproduce known knowledge

Concept of Learning

Machine Learning

- Different cases of machine learning
  - Supervised learning: inputs and corresponding outputs are bind together
  - Unsupervised learning: only inputs are available for the learner
  - Reinforcement learning: learning based on reward or punishment

Concept of Learning



Machine Learning Framework

Concept of Learning

Machine Learning Framework

- Environment
  - It refers to the nature and quality of the information given to the learning element
  - The nature of information depends on its level
    - High level: information is abstract and deals with broad class of problems
    - Low level: information is detailed and deals with single problem
  - Quality of information involves noise free, ordered and reliable information

Concept of Learning

Machine Learning Framework

- Learning Element
  - Acquire new knowledge through learning element
  - Learning may be of any type discussed above
- Problem Generator/ Knowledge Base
  - Stores the information about the problems and solutions are suggested
  - Knowledge Base should be
    - Expressive: Knowledge should be represented in easy and understandable way
    - Modifiable: Must be easy to update or add new data in the knowledge base
    - Extendibility: the knowledge base should have feature to change its structure that should be well defined

Concept of Learning

Machine Learning Framework

- Performance Element
  - This part analyses how complex the learning is and how learning is being performed
  - Complexity depends on the type of task to be performed
  - It must send feedback to the learning system as well so that evaluation of overall performance could be done
  - The learning element should have access to all internal actions of the performance elements
- Sensors and Actuators
  - Sensors collect information from environment
  - Actuators implement the suggested action

Concept of Learning

Winston's Learning or Semantic Nets Learning

- This program operated in simple blocks world domain
- Goal is to construct representations of the definitions of concepts in the block domain
- Eg. Learns the concepts of house, tents, and arches.
- Introduces near miss for each concept.
- Near miss is an object that is not an instance of the concept in question but that is very similar to such instances.

Concept of Learning

Winston's Learning or Semantic Nets Learning

- Begins with a structural description of one known instance of the concept
- Examine descriptions of other known instances of the concept
- Examine descriptions of near misses of the concept and restrict the definition to exclude this
- For example: refer to page number 459-461 of Artificial Intelligence, Rich and Knight

Learning by Analogy

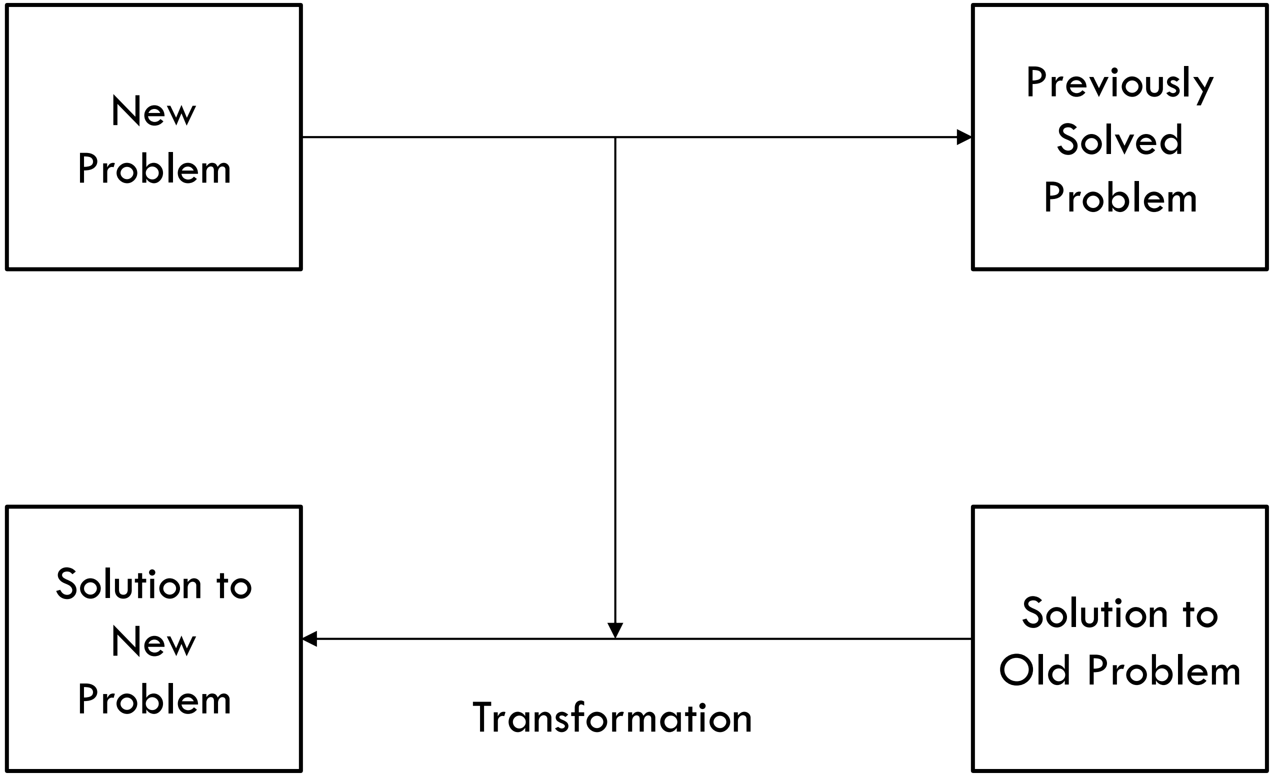
- Analogy is a powerful inference tool
- Our language and reasoning are laden with analogy
- Example:
  - Last month share market was a roller coaster.
  - Bill is like a fire engine
- So, AI must be able to grasp analogy for learning easily
- It is used in different learning strategies like learning by advice taking, learning in problem solving, etc.

Learning by Analogy

- Two methods of Analogy Problem solving are:
  - Transformational Analogy : focuses on final solution
  - Derivational Analogy: focuses on process of problem solving

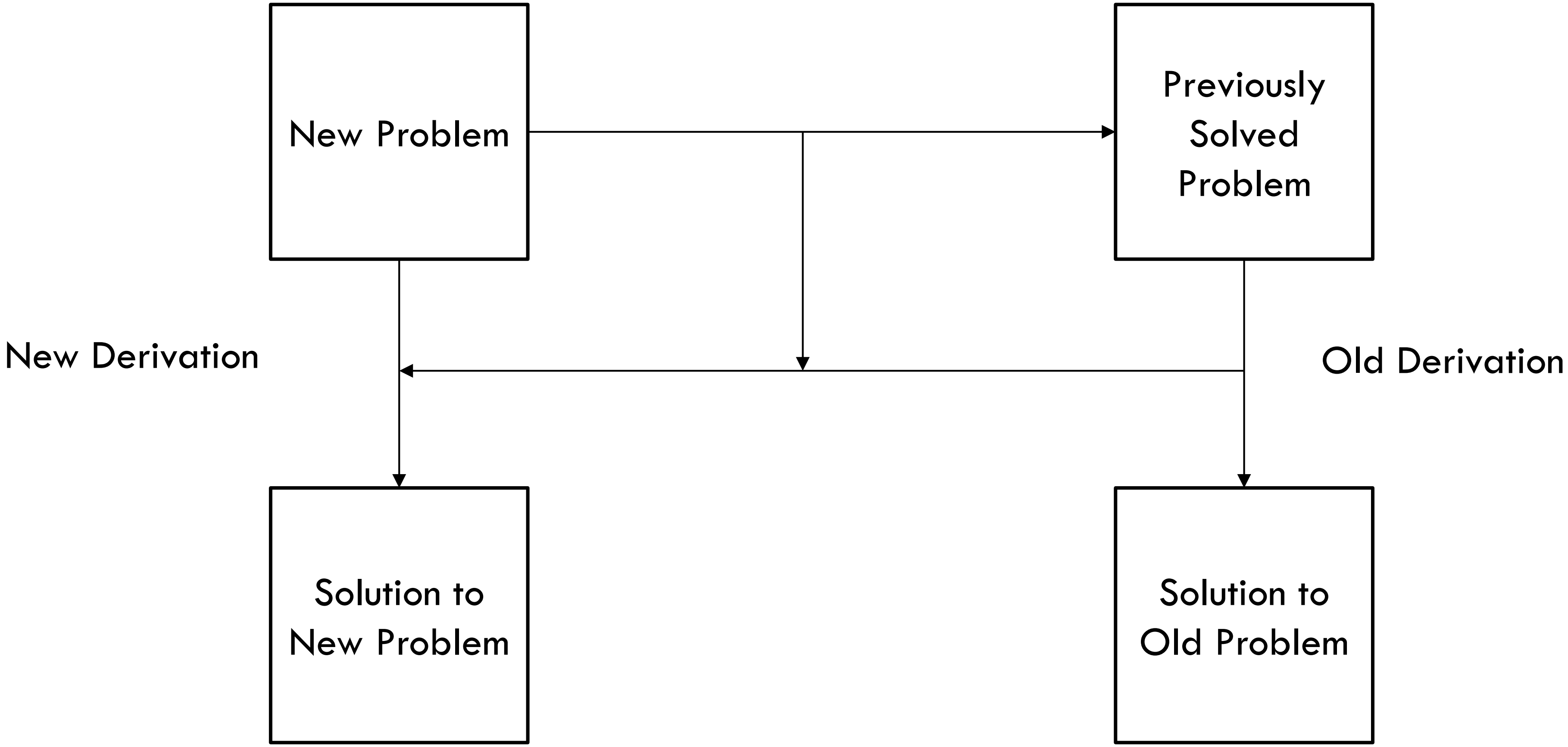
Learning by Analogy

Transformational Analogy



Learning by Analogy

Derivational Analogy



Inductive Bias Learning

- Based on classification of problems prior to solving
- Based on induction of result
- Based on generic facts, rules and patterns
- Also called concept learning
- Approaches:
  - Winston's Learning Program
  - Version Space
  - Decision Tree

Genetic Algorithm

- **Algorithm:** An algorithm is a sequence of instructions to solve a problem. Most of the algorithms are static.

- **A Genetic Algorithm(GA)** is adaptive (dynamic) a model of machine learning algorithm that derives its behavior from a metaphor of some of the mechanisms of evolution in nature.

GA - Background

- On 1 July 1858, **Charles Darwin** , presented his **theory of evolution.** This day marks the beginning of a revolution in Biology.
- **Darwin**'s classical **theory of evolution** , together with Weismann's **theory of natural selection** and Mandel's concept of **genetics** , now represent the **neo-Darwinism**
- **Neo=Darwinism** is based on process of reproduction, mutation, competition and selection.

GA - Background

- Evolution can be seen as a process leading to the maintenance of a population's ability to survive and reproduce in a specific environment. This ability is called **evolutionary fitness.**
- Evolutionary fitness can also be viewed as a measure of the organism's ability to anticipate changes in its environment.
- The fitness, or the quantitative measure of the ability to predict environmental changes and respond adequately, can be considered as the quality that is optimised in natural life

GA - Evolutionary Computation

- Evolutionary Computation stimulates evolution on a computer. The result of such simulations is a sense of optimisation algorithms
- Optimisation iteratively improves the quality of solutions until an optimal, or near-optimal, solution is found
- The evolutionary approach is based on computational models of natural selection and genetics. We call them **evolutionary computation** ,an umbrella term that combines **genetic algorithms, evolution strategies** and **genetic programming**

GA - Simulation of Natural Evolution

- All methods of evolutionary computation simulate natural evolution by creating a population of individuals, evaluating their fitness, generating a new population through genetic operations, and repeating this process a number of times.
- We focus on **Genetic Algorithm** as most of the other algorithms can be viewed as variations of genetic algorithms.

Genetic Algorithm

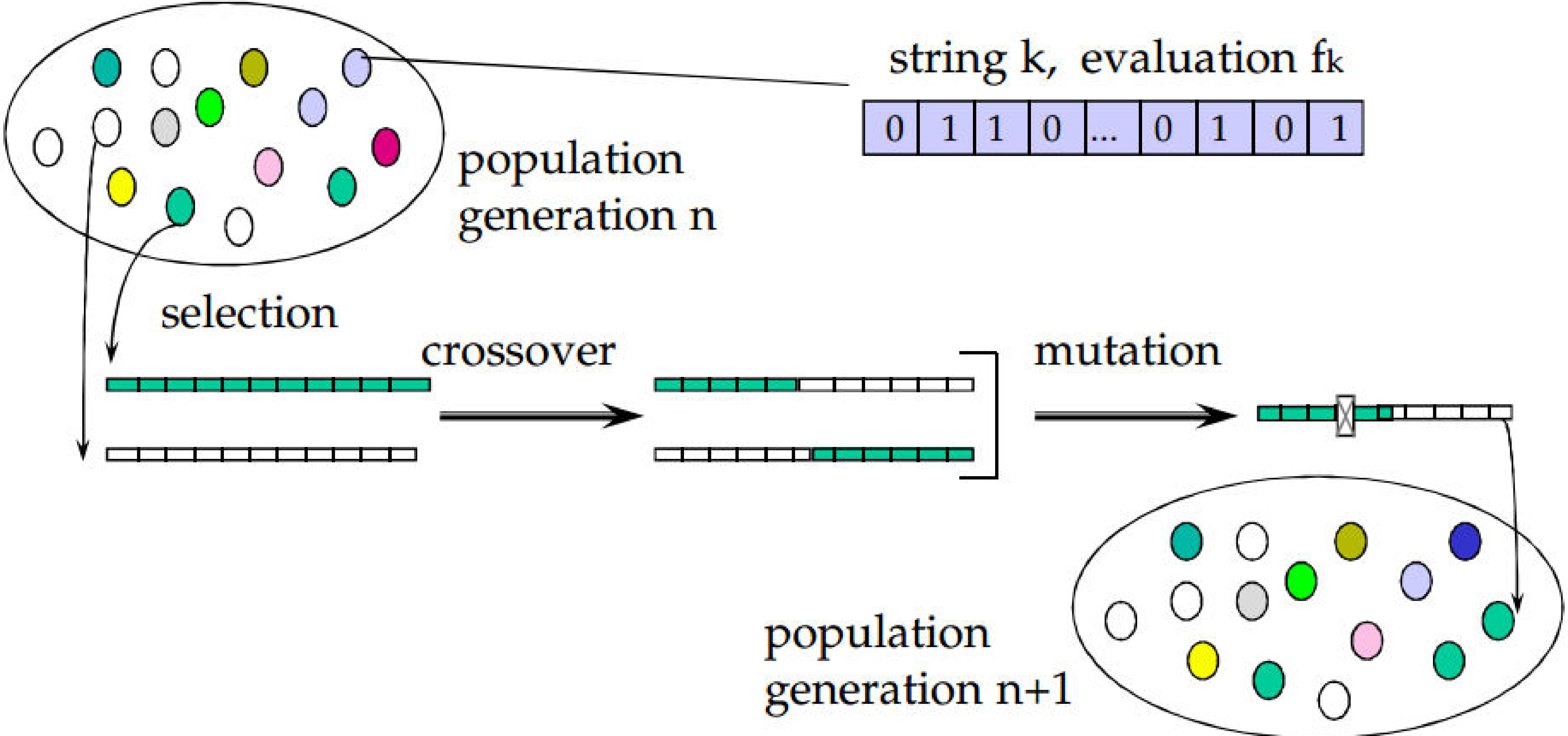
- In early 1970s John Holland introduced the concept of genetic algorithm
- His aim was to make computers do what nature does. Holland was concerned with algorithms that manipulate strings of binary digits
- Each artificial "chromosomes" consists of a number of "genes", and each gene is represented by 0 or 1

1	0	1	1	0	1	0	0	0	0	0	1	0	1	0	1

Genetic Algorithm

- Two mechanisms link a GA to the problem it is solving : **Encoding** and **Evaluation**
- The GA uses a measure of fitness of individual chromosomes to carry out reproduction. As reproduction takes place, the crossover operator exchanges part of two single chromosomes and he mutation operator changes the gene value in some randomly chosen location of the chromosome

Basic Genetic Algorithm



Genetic Algorithm

GA Operators and Parameters

- **Fitness function:** The fitness function is defined over the genetic representation and measures the *quality* of the represented solution.
- **Selection Operator:** Selects parents for reproduction based on relative fitness of candidates in the population
  - Roulette Wheel Selection
  - Ranking Selection

Genetic Algorithm

- **Crossover Operator:**
  - Exchanges part of chromosome between two parent chromosomes with some crossover rate(probability), typically 0.4 – 0.8
  - The main operator to provide exploitation in search building up good genes in chromosome
    - **One Point Crossover:** randomly chooses a crossover point where two parent chromosomes “break” and then exchanges the chromosome parts after that point. As a result, two new offspring is created
    - crossover points in two the chromosome parts spring are created.

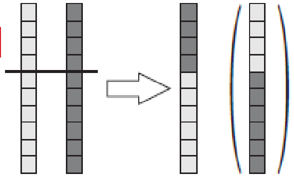


Fig. .a: Single-point Crossover (SPX).

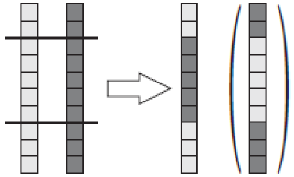


Fig. .b: Two-point Crossover (TPX).

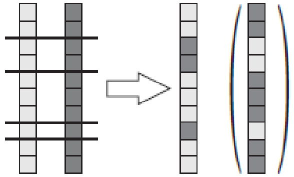
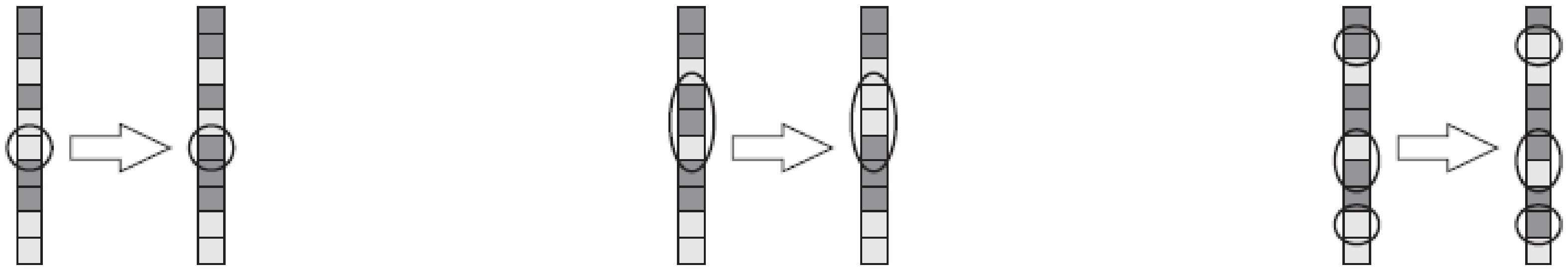


Fig. .c: Multi-point Crossover (MPX).

Genetic Algorithm

- **Mutation Operator:**
  - Changes a randomly selected gene in the chromosome
  - mimics random changes in genetic code
  - Background operator to provide exploration in search to avoid being trapped on a local optimum
  - Mutation probability is quiet small in nature and is kept low for GAs, typically in the range between [0.001 – 0.01] or



a: Single-gene mutation. b: Multi-gene mutation c: Multi-gene mutation

Genetic Algorithm

- **Elitism Approach :** Saves the best individual in next generation
- **Basic GA Parameters:**
  - Population size
  - Crossover rate (Probability)
  - Mutation Rate (Probability)
  - Number of Generation (a Stopping Criterion)

Steps in Genetic Algorithm

1. Represent the problem variable as a chromosome of a fixed length, choose the size of a chromosome population *N*, the crossover probability *pc* and the mutation probability *pm*.
2. Define a fitness function to measure the fitness of an individual chromosome in the problem domain.
3. Randomly generate an initial population of chromosomes of size *N*: $x1, x2, .., xN$
4. Calculate the fitness of each individual chromosome: $f(x1), f(x2), \ldots, f(xN)$
5. Select a pair of chromosomes for mating from the current population based on their fitness.
6. Create a pair of offspring chromosomes by applying the genetic operators – **crossover** and **mutation.**
7. Place the created offspring chromosomes in the new population.
8. Repeat *Step 5* until the size of the new chromosome population becomes equal to the size of the initial population, *N*.
9. Replace the initial (parent) chromosome population with the new (offspring) population.
10. Go to *Step 4*, and repeat the process until the termination criterion is satisfied.

Genetic Algorithm : Case Study

Example of Selection

Evolutionary Algorithms is to maximize the function $f(x) = x^2$ with x in the integer interval $[0, 31]$, i.e., $x = 0, 1, \ldots 30, 31$.

1. The first step is encoding of chromosomes; use binary representation for integers; 5-bits are used to represent integers up to 31.
2. Assume that the population size is 4.
3. Generate initial population at random. They are chromosomes or genotypes; e.g., 01101, 11000, 01000, 10011.
4. Calculate fitness value for each individual.
   (a) Decode the individual into an integer (called phenotypes),
   $01101 \rightarrow 13$; $11000 \rightarrow 24$; $01000 \rightarrow 8$; $10011 \rightarrow 19$;
   (b) Evaluate the fitness according to $f(x) = x^2$,
   $13 \rightarrow 169$; $24 \rightarrow 576$; $8 \rightarrow 64$; $19 \rightarrow 361$.
5. Select parents (two individuals) for crossover based on their fitness in p_i. Out of many methods for selecting the best chromosomes, if **roulette-wheel** selection is used, then the probability of the i^{th} string in the population is $p_i = F_i \,/\, (\sum_{j=1}^{n} F_j)$, where

   F_i is fitness for the string i in the population, expressed as $f(x)$

   p_i is probability of the string i being selected,

   n is no of individuals in the population, is population size, $n=4$

   $n * p_i$ is expected count

Genetic Algorithm : Case Study

String No	Initial Population	X value	Fitness Fi $f(x) = x^2$	p i	Expected count N * Prob *i*
1	0 1 1 0 1	13	169	0.14	0.58
2	1 1 0 0 0	24	576	0.49	1.97
3	0 1 0 0 0	8	64	0.06	0.22
4	1 0 0 1 1	19	361	0.31	1.23
Sum			1170	1.00	4.00
Average			293	0.25	1.00
Max			576	0.49	1.97

The string no 2 has maximum chance of selection.

6. Produce a new generation of solutions by picking from the existing pool of solutions with a preference for solutions which are better suited than others:

   We divide the range into four bins, sized according to the relative fitness of the solutions which they represent.

Strings	*Prob i*	*Associated Bin*
0 1 1 0 1	0.14	0.0 . . . 0.14
1 1 0 0 0	0.49	0.14 . . . 0.63
0 1 0 0 0	0.06	0.63 . . . 0.69
1 0 0 1 1	0.31	0.69 . . . 1.00

   By generating 4 uniform (0, 1) random values and seeing which bin they fall into we pick the four strings that will form the basis for the next generation.

Random No	*Falls into bin*	*Chosen string*
0.08	0.0 . . . 0.14	0 1 1 0 1
0.24	0.14 . . . 0.63	1 1 0 0 0
0.52	0.14 . . . 0.63	1 1 0 0 0
0.87	0.69 . . . 1.00	1 0 0 1 1

Genetic Algorithm : Case Study

7. Randomly pair the members of the new generation

Random number generator decides for us to mate the first two strings together and the second two strings together.

8. Within each pair swap parts of the members solutions to create offspring which are a mixture of the parents :

For the first pair of strings: **0 1 1 0 1 , 1 1 0 0 0**

– We randomly select the crossover point to be after the fourth digit. Crossing these two strings at that point yields:

0 1 1 0 1 ⇒ 0 1 1 0 |1 ⇒ 0 1 1 0 0

1 1 0 0 0 ⇒ 1 1 0 0 |0 ⇒ 1 1 0 0 1

For the second pair of strings: **1 1 0 0 0 , 1 0 0 1 1**

– We randomly select the crossover point to be after the second digit. Crossing these two strings at that point yields:

1 1 0 0 0 ⇒ 1 1 |0 0 0 ⇒ 1 1 0 1 1

1 0 0 1 1 ⇒ 1 0 |0 1 1 ⇒ 1 0 0 0 0

Genetic Algorithm : Case Study

9. Randomly mutate a very small fraction of genes in the population :
   With a typical mutation probability of per bit it happens that none of the bits in our population are mutated.

10. Go back and re-evaluate fitness of the population (new generation) :
    This would be the first step in generating a new generation of solutions. However it is also useful in showing the way that a single iteration of the genetic algorithm has improved this sample.

String No	Initial Population (chromosome)	X value (Pheno types)	Fitness $f(x) = x^2$	Prob i (fraction of total)	Expected count
1	0 1 1 0 0	12	144	0.082	0.328
2	1 1 0 0 1	25	625	0.356	1.424
3	1 1 0 1 1	27	729	0.415	1.660
4	1 0 0 0 0	16	256	0.145	0.580
Total (sum)			1754	1.000	4.000
Average			439	0.250	1.000
Max			729	0.415	1.660

Observe that :

1. Initial populations : At start step 5 were

   0 1 1 0 1 , 1 1 0 0 0 , 0 1 0 0 0 , 1 0 0 1 1

   After one cycle, new populations, at step 10 to act as initial population

   0 1 1 0 0 , 1 1 0 0 1 , 1 1 0 11 , 1 0 0 0 0

2. The total fitness has gone from **1170** to **1754** in a single generation.
3. The algorithm has already come up with the string 11011 (i.e $x = 27$) as a possible solution.

Explanation Based Learning

- *Explanation-based learning (EBL)* uses a domain theory to construct an explanation of the training example, usually a proof that the example logically follows from the theory
- Using this proof the system filters the noise, selects only the relevant to the proof aspects of the domain theory, and organizes the training data into a systematic structure
- This makes the system more efficient in later attempts to deal with the same or similar examples.

Explanation Based Learning

Basic Concept of EBL

- *Target concept*. The task of the learning system is to find an effective definition of this concept. Depending on the specific application the target concept could be a classification, theorem to be proven, a plan for achieving goal, or heuristic to make a problem solver more efficient.
- *Training example*. This is an instance of the target concept
- *Domain theory*. Usually this is a set of rules and facts representing domain knowledge. They are used to explain how the training example is an instance of the target concept.
- *Operationality criteria*. Some means to specify the form of the concept definition.

Boltzmann Machine

- It’s a network of symmetrically connected, neuron like units that make stochastic decision about whether to be on or off
- Has simple learning algorithm that allows it to discover interesting features that represent complex regularities in training data
- Very slow learning
- For searching, Boltzmann machine has fixed weights on the connection but for learning weights use small update

Boltzmann Machine

- Output y is given by,

$$y = b_i + \sum x_j w_{ij}$$

- Where x_i=1 for on state and 0 for off
- Probability of being on is

$$Prob(x_j = 1) = \frac{1}{1 + e^{-y}}$$

Thank You