Genetic Algorithm

Introduction

This algorithm was proposed by Holland in the year 1975. This algorithm is suited for solving complex optimization problems. They are search algorithms based on the theory of natural genetics i.e. operations existing in nature. The advantage in this algorithm is that they search complex and large amount of spaces efficiently and locate near optimal solution. Genetic algorithms operate on asset of individual elements called population. There is a set of biologically inspired operators that can change these individuals. According to the evolutionary theory, only the more suited individuals in the population are likely to survive and to generate offspring, thus transmitting their biological heredity to other generations.
In computing terms, GA maps strings of numbers to each potential solution. Each solution becomes an individual in the population and each string becomes a representation of an individual. We should be able to derive individual from its string representation. The GA then manipulates these strings in search of improved solution.

Five phases are considered in a genetic algorithm.

1. Initial population
2. Fitness function
3. Selection
4. Crossover
5. Mutation

GA Process: The steps involved in GA are

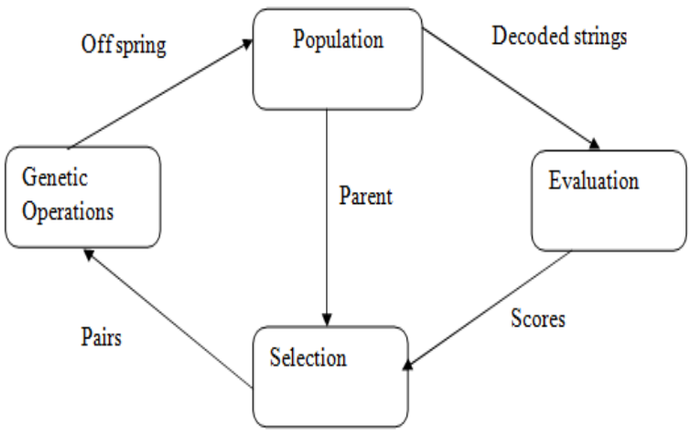


Figure 8.6 Reproduction Cycle

- Creation of population strings
- Evaluation of each string
- Selection of the best string.
- Genetic operations to create new population of strings.

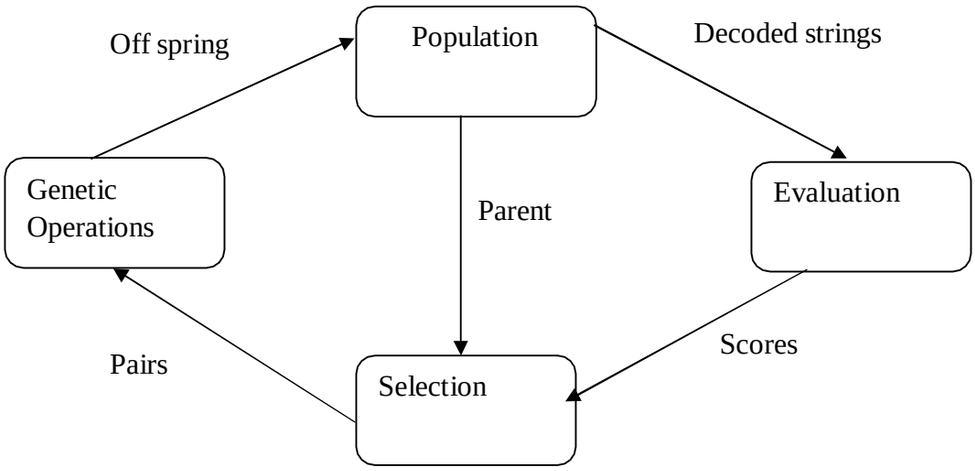
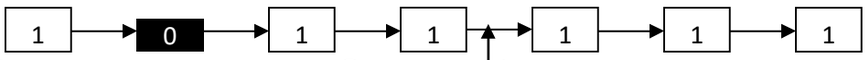


Figure 8.6 Reproduction Cycle

Each cycle produces a new generation of possible solutions to a problem. At the first stage a population of the possible solution is created at the starting point. Each individual in the population is encoded to a string to be manipulated by genetic operators. In the next stage, each individual is evaluated. Initially, the individual is created from its string description, and then its performance in relation to the target is evaluated.

Cross Over: It is one of the genetic operators used to recombine the population genetic material. It takes two chromosomes and swaps a part of their genetic information to produce new chromosomes. The cross over point is randomly chosen from the parent chromosomes. Parent1 and parent2 combine to generate a new offspring, son. This process results in a special genetic structure called building blocks that are retained for future generations.

Parent1:



Parent2: Cross over point

Son:

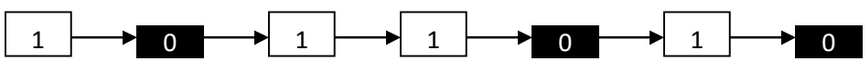


Figure 8.7 Cross Over

Mutation: This operator introduces new genetic structures in the population by randomly changing some of its building blocks. The modification done here is totally random and is not related to any previous genetic structures. The mutation process is implemented by occasionally altering a random bit from a chromosome.

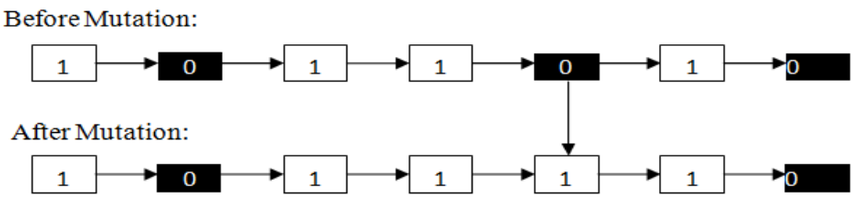


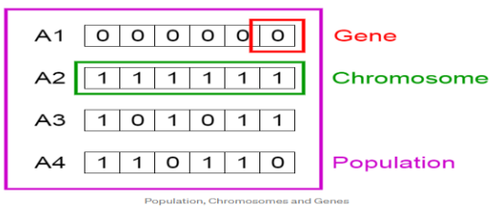
Figure 8.8 Mutation

Initial Population

The process begins with a set of individuals which is called a **Population**. Each individual is a solution to the problem you want to solve.

An individual is characterized by a set of parameters (variables) known as **Genes**. Genes are joined into a string to form a **Chromosome** (solution).

In a genetic algorithm, the set of genes of an individual is represented using a string, in terms of an alphabet. Usually, binary values are used (string of 1s and 0s). We say that we encode the genes in a chromosome.



Population, Chromosomes and Genes

Fitness Function

The **fitness function** determines how fit an individual is (the ability of an individual to compete with other individuals). It gives a **fitness score** to each individual. The probability that an individual will be selected for reproduction is based on its fitness score.

Selection:

The idea of **selection** phase is to select the fittest individuals and let them pass their genes to the next generation.

Two pairs of individuals (**parents**) are selected based on their fitness scores. Individuals with high fitness have more chance to be selected for reproduction.

Why use Genetic Algorithms

- They are Robust
- Provide optimisation over large space state.
- Unlike traditional AI, they do not break on slight change in input or presence of noise

Application of Genetic Algorithms

Genetic algorithms have many applications, some of them are –

- Recurrent Neural Network
- Mutation testing
- Code breaking
- Filtering and signal processing
- Learning fuzzy rule base etc

Assignment

Short Answer Questions

1. What is mutation?
2. Define genetic algorithm.
3. List the four steps involved in Genetic algorithm.
4. Explain cross over in genetic algorithm.
5. What isselection?

Long Answer Questions

1. Explain mutation and crossover operation in genetic algorithm.(5 Marks-M.U.April/May 2012, 2010)
2. Describe the technique of Genetic algorithm. 5 Marks-M.U. April/May 2011,2009)
3. Write a short note on genetic algorithm.
4. Explain fitness and selection operation.

Clustering technique

Introduction

Clustering is useful technique for discovery of data distributions and patterns of under lying data. The goal of clustering is to discover both the dense and sparse regions in a data set. The main emphasis has been to cluster with as high accuracy as possible, while keeping the I/O costs high. Data clustering has been studied in the statistics, machine learning, and database communities with diverse emphases. It is also to be noted that the basic principle of clustering hinges on a concept of distance metric or similarity metric. Since the data are invariably real numbers for statistical applications and pattern recognition, a large class of metrics exists and one can define one's own metric depending on a specific requirement. But in databases, there exist objects which cannot be ordered, and their attributes can be categorical in nature. For example, consider a market-basket database. Typically, the number of items, and thus the number of attributes in such a database is very large, while the size of an average transaction is much smaller. Furthermore, customers with similar buying patterns, who belong to a single cluster, may buy a small subset of items from a much larger set that defines the cluster. Thus, conventional clustering methods that handle only numeric data are not suitable for data mining purposes.

Clustering Paradigms

There are two main approaches to clustering-hierarchical clustering and partitioning clustering. Besides, clustering algorithms differ among themselves in their ability to handle different types of attributes, numeric and categorical, in accuracy of clustering, and in their ability to handle disk-resident data.

Hierarchical VS Partitioning

The partition clustering techniques partition the database into a predefined number of clusters. They attempt to determine k partitions that optimize a certain criterion function. The partition clustering algorithms are of two types: k-means algorithms and k-medoid algorithms. Another type of algorithms can be k-mode algorithms.

The hierarchical clustering techniques do a sequence of partitions, in which each partition is nested into the next partition in the sequence. It creates a hierarchy of clusters from small to big or big to small. The hierarchical techniques are of two types-agglomerative and divisive clustering techniques. Agglomerative clustering techniques start with as many clusters as there are records, with each cluster having only one record. Then pairs of clusters are successively merged until the numbers of clusters reduces to k. At each stage, the pairs of the clusters that are merged are the ones nearest to each other. If the merging is continued, it terminates in a hierarchy of clusters which is built with just a single cluster containing al1 the records, at the top of the hierarchy. Divisive clustering techniques take the opposite approach from agglomerative techniques. These starts with all the records in one cluster, and then try to split that cluster into small pieces.

Numeric VS Categorical

Clustering can be performed on both numerical data and categorical data. For the clustering of numerical data, the inherent geometric properties can be used to define the distances between the

points. But for clustering of categorical data, such a criterion does not exist and many data sets also consist of categorical attributes, on which distance functions are not naturally defined.

Partitioning Algorithms

This algorithm constructs partition of database with N object into set of k clusters. The constructors involve determining the optimal partitioning with respect to an objective function. The partition algorithm is implemented in two approaches

a) Exhaustive enumeration
b) Iterative optimization

In exhaustive enumeration we can find optimal partitioning but practically it is infeasible accept when N and k are small. Iterative optimization paradigm starts with the initial partition and uses an iterative control strategy. It tries swapping data points to see if such a swapping improves the quality of clustering. When swapping does not yield any improvements in clustering, it finds a locally optimal partition. This quality of clustering is very sensitive to the initially selected partition. The various algorithms under partition clustering are

i. k-means algorithm
ii. k-medoid algorithm

In k-means algorithm each cluster is represented by center of gravity of the cluster. In k-medoid algorithm each cluster is represented by one of the objects of cluster located near the center. Most of special clustering algorithms designed for data mining are k-medoid algorithms.

K-medoid Algorithms

PAM: (Partition around Medoids) uses a k-medoid method to identify the clusters. PAM selects *k* objects arbitrarily from the data as medoids. Each of these *k* objects is representatives of *k* classes. Other objects in the database are classified based on their distances to these k-medoids (we say that the database is partitioned with respect to the selected set of k-medoids). The algorithm starts with arbitrarily selected k-medoids and iteratively improves upon this selection.

In each step, a swap between a selected object O_i and non selected object O_h is made, as long as such a swap results in an improvement in the quality of clustering. To calculate the effect of such a swap between O_i; and O_h a cost C_{ih} computed, which is related to the quality of partitioning the non-selected objects to *k* clusters represented by the medoids.

Partitioning: If O_j is a non-selected object and 0_i is a medoid, we then say that q belongs to the cluster represented by O_i if $d(0_i , O_j) = \text{Min}O_e\ d(O_j, O_e)$,where the minimum is taken over all medoids O_e and $d(0_a , O_b)$ determines the distance, or dissimilarity, between objects O_a and O_b. The dissimilarity matrix is known prior to the commencement of PAM. The quality of clustering is measured by the average dissimilarity between an object and the medoid of the cluster to which the object belongs

CLARA

It can be observed that the major computational efforts for PAM are to determine k-medoids through an iterative optimization. Though CLARA follows the same principle, it attempts to reduce the computational effort. Instead of finding representative objects for the entire data set, CLARA draws a sample of the data set, and applies PAM on this sample to determine the optimal set of medoids from the sample. It then classifies the remaining objects using the partitioning principle. If the sample were drawn in a sufficiently random way, the medoids of the

sample would approximate the medoids of the entire data set. The steps of CLARA are summarized below.

CLARA Algorithm
Input: Database of D objects.
repeat for m times

draw a sample $S \subseteq D$ randomly from D.
call PAM (S, k) to get k medoids.
classify the entire data set D to C_1, C_2 ...C_K.
calculate the quality of clustering as the average dissimilarity.

end.

Hierarchical clustering:

These techniques do a sequence of partitions, in which each partition is nested into the next partition in the sequence. Thus it creates the hierarchy of clusters from big to small or small to big.

The different types of hierarchical clustering are

a) Agglomerative
b) Divisive

Agglomerative: Agglomerative approach it starts with many cluster where each cluster consist of one record. The pairs of cluster are successively merged until the number of cluster reduces to k. At each stage the pairs of cluster are merged to the ones nearest to each other.

Divisive: Divisive clustering is the opposite approach from agglomerative clustering. It starts with all records in one cluster and then split the cluster into smaller pieces depending on the data that is used for clustering.

We can classify divisive clustering into two types

a) Numerical
b) Categorical

For the clustering of numerical data, the inherent geometric properties can be used to define the distances between the points. But for clustering of categorical data, such a criterion does not exist and many data sets also consist of categorical attributes, on which distance functions are not naturally defined.

DBSCAN

DBSCAN (Density Based Spatial Clustering of Applications of Noise) uses a density based notion of clusters to discover clusters of arbitrary shapes. The key idea of DBSCAN is that, for each object of a cluster, the neighbourhood of a given radius has to contain at least a minimum number of data objects. In other words, the density of the neighbourhood must exceed a threshold. The critical parameter here is the distance function for the data objects.

ε-Neighbourhood of an Object: For a given non-negative value ε, the ε -neighbourhood of an object O_i denoted by $N\ \varepsilon\ (O_i)$, is defined by $N\ \varepsilon\ (O_i) = \{\ O_j \in D \mid d(O_i, O_j) \leq \varepsilon\ \}$

Core Object: An object is said to be a Core Object if $| N\ \varepsilon\ (O) | \geq$ MinPts. A core object is an object which has a neighbourhood of user-specified minimum density.

Directly-density-reachable: An object O_i is directly-density-reachable from an object O_j with respect to ε and MinPts, if O_j is a core object and O_i is in its ε -neighbourhood.

1. $O_i \in N\ \varepsilon\ (O_j)$
2. $|N\ \varepsilon\ (O_j)| \geq$ MinPts. (In other words O_j is a core object)

Density-reachable: An object O_i is density-reachable from an object O_j with respect to ε and MinPts in D if there is a chain of objects 0_1, $O_2 \ldots O_n$ such that $O_i = O_j$ $O_n = O_i$ such that $O_e \in D$ and O_{e+1} is directly-density-reachable from Oe with respect ε and MinPts in D

Density-connected: An object O_i, is density-connected to an object O_j with respect to ε and MinPts in D if there is another object $0 \in D$, such that both O_i and 0_j are density-reachable from 0, with respect to ε and MinPts in D.

Cluster: A Cluster C with respect to ε and MinPts is a non-empty subset of D satisfying the following condition

For all O_i $O_j \in$ N, if O_i, $\in$ C and O_i; is density-reachable from O_j, with respect to & and MinPts, then $O_i \in C$

For all O_i $O_{j,} \in$ C, is density connected to O_j with respect to ε and MinPts.

Noise: Let C_1, C_2, .., Ck be the clusters of D with respect to ε and MinPts. Then, we define the noise as the set of objects in D which do not belongs to any cluster C_i as Noise $\{O \in D \mid \forall\ i, O \notin C_i\}$. The noise as the set of objects in D which do not belong to any cluster C, as Noise

We identify two different kinds of objects in a DBSCAN-core objects and non- core objects. Non-core objects, in turn, are either border objects or noise objects. Two border objects are possibly not density-reachable from each other. However, a border object is always density-reachable' from a core object. A noise object is a non-core object, which is not density-reachable from other core objects. DBSCAN's procedure, for finding a cluster is based on the fact that a cluster is uniquely determined by any of its core objects.

The DBSCAN algorithm maintains the set of objects in three different categories. These are: classified, unclassified and noise. Each classified objects has an associated c1uster-id, indicating the cluster in which it is included. A noise object may also have an associated dummy cluster-id. For both classified and noise objects, the ε -neighbourhoods are already computed. The unclassified category of objects does not have any cluster-id and their neighbourhoods are not computed. The algorithm gradually converts an unclassified object into a classified or a noise object.

Categorical Clustering Algorithm

The important Categorical Clustering algorithms are STIRR, ROCK, and CACTUS. One interesting common feature of these three algorithms is that they attempt to model the similarity of categorical attributes in more or less a similar manner. A and B are deemed to be similar if the sets of items with which they co-occur have a large overlap, even though these never co-occur together. ROCK tries to introduce a concept called neighbours and link. Two items have a link if they have a common neighbor. STIRR defines a weighted node and the weights of co-occurring

items are propagated and, as a result, two items sharing common co-occurring items will exhibit a similar magnitude of weights. CACTUS also makes use of occurrences as the similarity measure. One can even go one step further to say that if A and B are similar and Band C are deemed similar, then we should be able to infer a weak type of similarity between A and C. In that way, similarity may uncover more distant co-relations. In the following three sections, we shall briefly outline these three algorithms.

STIRR

STIRR (Sieving through Iterated Relational Reinforcement), proposed by Gibson, Kleinberg and Raghavan, is an iterative algorithm based on non-linear dynamical systems. The salient features of STIRR are the following.

The database is represented as a graph, where each distinct value in the domain of each attribute is represented by a weighted node. Thus, if there are N attributes and the domain size of the I^{th} attribute is d_i, then the number of nodes in the graph is $\sum d_i$. For each tuple in the database, an edge represents a set of nodes which participate in that tuple. Thus, a tuple is represented as a collection of nodes, one from each attribute type. We assign a weight to each node. The set of weights of all the nodes define the configuration of this structure. The algorithm proceeds iteratively to update the weight of each node, based on the weights of other nodes to which it is connected. Thus, it moves from one configuration to the other till it reaches a stable point. The updating of the weights depends on a combiner function that combines the weights of the nodes participating in a given tuple. Their convergence is dependent on the combiner function.

Initial configuration: The initial weights of nodes can be either assigned uniformly, or randomly, or by a focussing technique. In the uniform initialization, all weights are set to 1. In the random initialization, each weight is an independently selected random value in the interval [0,1]. We can focus a portion of the graph by initializing the portion to 1 and the remaining part of the graph to 0.

Weight Update: STIRR iteratively changes the configuration by updating the weight of any single node. The new weight of any node is calculated, based on a combiner function. Typically, a combiner function combines the weights of other nodes participating in any tuple with the given node for which the weight is to be updated. A combiner function can simply add all the weights, or can multiply all the weights, or may combine them in some other way.

Assignment-5

Short Answer Questions (2 Marks each)

1. Define clustering. (M.U. April/May 2008, 2009, 2010, 2011, 2012)
2. Compare Hierarchical and Partitioning clustering. (M.U. April/May 2011)
3. Differentiate Numerical and categorical clustering.
4. What are the two approaches in order to implement partitioning algorithm
5. What are the two types of algorithm under partitioning clustering?
6. What is PAM?
7. Why is it difficult to handle categorical data for clustering? (M.U. April/May 2011)
8. Compare agglomerative clustering with divisive clustering.
9. Expand DBSCAN? Why it is used?
10. Define ε-Neighbourhood of an Object?
11. What is core object?

12. Define density reachable.
13. Define density connected.
14. What is noise in clustering?
15. List out different types of categorical clustering.
16. What is STIRR?

Long Answer Questions

1. Differentiate hierarchical and partitioning clustering. (5 Marks- M.U. April/May 2012)
2. What is numerical clustering? How it is different from categorical clustering? (5 Marks- M.U. April/May 2012, 2010)
3. Compare agglomerative clustering with divisive clustering. (5 Marks- M.U. April/May 2010, M.U. Oct./Nov.2013)
4. Explain categorical clustering. (5 Marks- M.U. April/May 2008)
5. Write a short note on partioning clustering. (5 Marks)
6. Write a short note on k-medoid algorithm. (5 Marks)
7. Compare categorical and numerical clustering. (5 Marks, M.U. Oct./Nov. 2013)
8. Write a note on CLARA. (4 Marks, M.U. Oct./Nov. 2013)
9. Write a short note on hierarchical algorithm. (5 Marks)
10. Explain DBSCAN algorithm. (10 Marks)
11. Write a note on STI RR. (4 Marks, M.U. Oct./Nov. 2013)

Datamining