Detection of Outlier from Large Data Sets Using Genetic Algorithm

G.Shoba[1] (Senior Assistant Professor)
Department of Computer Science and Engineering
CCET (affiliated to Pondicherry University)
Puducherry, India

M.Rajeswari[2] (M.Tech-Final Year)
Department of Computer Science and Engineering
CCET (affiliated to Pondicherry University)
Puducherry, India

S.Kalaitchelvi[3] (M.Tech-Final Year)
Department of Computer Science and Engineering
CCET (affiliated to Pondicherry University)
Puducherry, India

Abstract - Outliers in a dataset is defined in a relaxed way as an observation that is significantly dissimilar from the residue as if it is produced by different mechanism which are outstanding from the remaining data in a dataset. Discovering outliers is an important issue in many of the common applications like fraud recognition, invasion detection, medicine, network sturdiness analysis and so on. Finding the Outliers or the odd instances will be more exciting compared to identifying the familiar data of common form. In this paper we proposed a Genetic algorithm to detect outliers from large data sets. The main objective the outlier detection is to find the data that are exceptional from other data in the data set.

Key Terms
Data Mining, Outlier, Outlier Detection, Genetic Algorithm.

I. INTRODUCTION

Data mining

Larger and larger amounts of data are collected and stored in data bases. This increases need of efficient and effective analysis methods to make use of the information contained implicitly in the data. Knowledge discovery in databases (KDD) has been defined as the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable knowledge from the data. Most studies in KDD focus on finding common patterns (association mining), grouping patterns (clustering), etc. However, for applications such as detecting criminal

activities of various kinds (e.g. in electronic commerce), rare events, deviations from the majority, or exceptional cases may be more interesting and useful than the common cases. "The data objects that do not comply with the general behavior or model of the data" and such data objects, which are grossly different from or inconsistent with the remaining set of data, are called outliers [1].

Outlier detection

Outlier detection has attracted increasing attention in machine learning, data mining and statistics literature. Outliers always refer to the data objects that are markedly different from or inconsistent with the normal existing data [1], [2]. A well known definition of "outlier" is given in [2]: "an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism," which gives the general idea of an outlier and motivates many anomaly detection methods [5]. Practically, outlier detection has been found in wide-ranging applications from fraud detection for credit cards, insurance or health care, intrusion detection for cyber-security, fault detection in safety critical systems, to military surveillance [1].In many of the data mining applications identifying the outliers or rare events discovers some new interesting and unexpected knowledge in many areas. It has been examined that in most of the algorithms that are developed to detect anomaly are not accurate [2]. It may detect the false data or an additional data which are not outliers which leads to false result. The results thus produced are also not optimized.

The remainder part of this paper is described as follows: In Section 2 we discussed about the related work done and the proposed work in detail and Section 3 describes genetic algorithm in detail and. Finally Section 4 concludes the paper with future work.

II. RELATED WORK

Various Approaches to Detect Outliers in Data Mining

In the past, many outlier detection methods have been proposed [1]. Typically, these existing approaches can be divided into four categories: distribution (statistical)-based clustering based, density-based and Distance based approaches.

Statistical based approach

Statistical approaches assume that the data follows some standard or predetermined distributions, and this type of approach aims to find the outliers which deviate from such distributions. The methods in this category always assume the normal example follow certain of data distribution. Nevertheless, we cannot always have this kind of priori data distribution knowledge in practice, especially for high dimensional real data sets. [1].

Distribution based approach (Rosseeuw 1996) had developed statistical methods from the given data and applied statistical test to find the object belong to a particular model or not. The

objects with low probability are identified as outliers in the statistical model. Because the distribution based approaches are univariate in nature they cannot be applied in multidimensional data space.

Clustering based approach

For clustering-based approaches , they always conduct clustering-based techniques on the samples of data to characterize the local data behavior. In general, the sub-clusters contain significantly less data points than other clusters, are considered as outliers. For example, clustering techniques has been used to find anomaly in the intrusion detection domain. In the work of [2], the clustering techniques iterative detect outliers to multi-dimensional data analysis in subspace. Since clustering-based approaches are unsupervised without requiring any labeled training data, the performance of unsupervised outlier detection is limited.

Density based approach

In addition, density-based, is one of the representatives of this type of approaches are local outlier factor (LOF) and variants [2]. Based on the local density of each data instance, the LOF determines the degree of outlierness, which provides suspicious ranking scores for all samples. The most important property of the LOF is the ability to estimate local data structure via density estimation. The advantage of these approaches is that they do not need to make any assumption for the generative distribution of the data. However, these approaches incur a high computational complexity in the testing phase, since they have to calculate the distance between each test instance and all the other instances to compute nearest neighbors.

Distance based approach

In the Distance based approach [(knorr 2000, angiulli 2005) the outliers are detected by a distance measure on the feature space. In ramasamy(2000), the outliers are identified by using k-nearest neighbor method to rank the outliers. The problem with this approach is that it is very difficult to find a particular value in a dataset [4].

Modes of outlier detection

Depending on the availability of a training dataset, outlier detection techniques described above operate in two different modes: supervised and unsupervised modes. Among the four types of outlier detection approaches, distribution-based approaches and model-based approaches fall into the category of supervised outlier detection, which assumes the availability of a training dataset that has labeled instances for normal class (as well as anomaly class sometimes). In addition, several techniques have been proposed that inject artificial anomalies into a normal dataset to obtain a labeled training data set. In addition, the work of presents a new method to detect outliers by utilizing the instability of the output of a classifier built on bootstrapped training data. Many algorithms have been proposed to identify the outliers but optimized solution has not been defined.

Generalized genetic algorithm

In this paper, we proposed a generalized genetic algorithm for identifying the exceptional objects from the dataset which also includes outliers. This is due to the fact that Genetic algorithm are very simple and easy to use and also computationally powerful. Many of the searching and optimization algorithms are not adaptive. In the sense that they generally solve only the given problem. Since the algorithm is designed for their problem alone. But Genetic algorithm are adaptive and robust in nature, they can be applied to any domain and to any type of problem with slight modifications in the representation, fitness value or with the choice of the genetic operators. But the behavior of the genetic algorithm remains same. So we had chosen Genetic algorithm as our algorithm to solve outliers. In our approach the outliers are identified based on the fitness value that is generated. The fitness values that are lower are considered to be outlier[3].

III.GENETIC ALGORITHM

GAs simulate the survival of the fittest among individuals over consecutive generation for solving a problem. Each generation consists of a population of character strings that are analogous to the chromosome that we see in our DNA. Each individual represents a point in a search space and a possible solution. The individuals in the population are then made to go through a process of evolution.

GAs are based on an analogy with the genetic structure and behaviour of chromosomes within a population of individuals using the following foundations:

- Individuals in a population compete for resources and mates.
- Those individuals most successful in each 'competition' will produce more offspring than those individuals that perform poorly.
- Genes from `good' individuals propagate throughout the population so that two good parents will sometimes produce offspring that are better than either parent.
- Thus each successive generation will become more suited to their environment.

Search Space

A population of individuals is maintained within search space for a GA, each representing a possible solution to a given problem. Each individual is coded as a finite length vector of components, or variables, in terms of some alphabet, usually the binary alphabet {0,1}. To continue the genetic analogy these individuals are likened to chromosomes and the variables are analogous to genes. Thus a chromosome (solution) is composed of several genes (variables).

A fitness score is assigned to each solution representing the abilities of an individual to `compete'. The individual with the optimal (or generally near optimal) fitness score is sought.

The GA aims to use selective `breeding' of the solutions to produce `offspring' better than the parents by combining information from the chromosomes.

The GA maintains a population of n chromosomes (solutions) with associated fitness values. Parents are selected to mate, on the basis of their fitness, producing offspring via a reproductive plan. Consequently highly fit solutions are given more opportunities to reproduce, so that offspring inherit characteristics from each parent. As parents mate and produce offspring, room must be made for the new arrivals since the population is kept at a static size. Individuals in the population die and are replaced by the new solutions, eventually creating a new generation once all mating opportunities in the old population have been exhausted. In this way it is hoped that over successive generations better solutions will thrive while the least fit solutions die out.

New generations of solutions are produced containing, on average, more good genes than a typical solution in a previous generation. Each successive generation will contain more good `partial solutions' than previous generations. Eventually, once the population has converged and is not producing offspring noticeably different from those in previous generations, the algorithm itself is said to have converged to a set of solutions to the problem at hand.

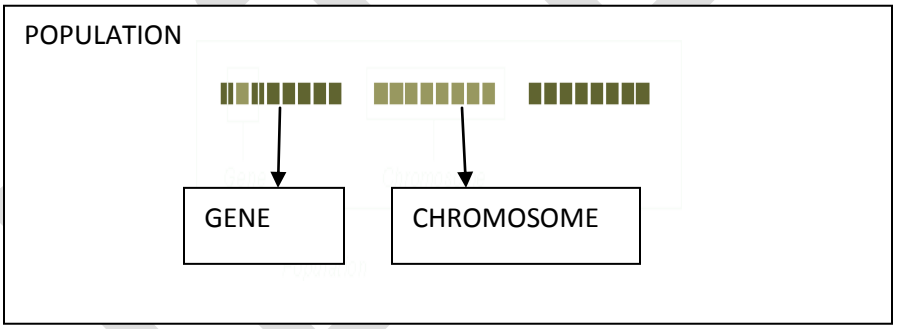

Fig 3.1: Search space for genetic algorithm

Compared to other searching algorithms Genetic algorithms are adaptive heuristic and robust in nature which implies that they can be applied problems of any domain with slight modification of the representation, fitness evaluation and the choice of the genetic operators but the basic operation of the algorithm remains same. Unlike other search algorithms, GA does not require any additional information about the problem.

In Genetic algorithm, the chromosomes are generated with a set of population strings which encode candidate solutions called individuals. Each individual led to an optimization problem which evolves toward better optimization solution. In general the solutions are represented as strings of 0's and 1's in the binary form, but other encodings like value, permutation encoding are also possible. Here, the evolution starts from the initial population with the randomly selected individuals. Then it seeks to optimize the function by a continuous

iterative process of selection, crossover and mutation until global optimum population is obtained.

In each generation, every individual's fitness is evaluated by selecting multiple individuals based on their fitness value from the current population and recombined to form new population. This obtained new population is used in the next iteration of the algorithm. The termination of the algorithm takes place when a maximum number of generations has been reached from the population.

The genetic algorithm employs the following process to produce the best individuals from the initial set of populations:

- **Input:** Set on N Chromosomes in the search space.
- **Output:** Outliers with lowest fitness value.
- **[Start]** Generate random population of N individuals.
- **[Fitness]** Fitness function f(x) for each chromosome is evaluated.
- **[New Population]** Repeat the following steps to create new population
    - **[Selection]** Select two parents from the population according to their fitness.
    - **[Crossover]** With the crossover probability crossover the parents to form new offspring. If no crossover is performed the offspring is resulted as parents.
    - **[Mutation]** With the mutation probability mutate the offspring at each locus.
    - **[Accept]** Place new offspring in the population.
- **[Replace]** Use new generated population for the next iteration.
- 5.**[Test]** If the termination condition is satisfied, return the best solution.
- 6.**[Result]**Sort the fitness value in descending order, the lower value are identified as outliers.
- 7.**[Loop]** Go to step 2 for next iteration.

Genetic Operators

In Genetic algorithm the representation of genes as chromosomes is done by encoding. The primary steps that are involved in the genetic algorithm are initialization, selection, reproduction (crossover and mutation) and termination.The first step in the genetic algorithm is the representation of the chromosomes in the problem space with suitable encoding techniques. Various techniques like binary encoding, value encoding and permutation encoding are available[3].

(i) Fitness function

In each problem the fitness function is formulated in way to solve it by genetic algorithm. A fitness function is a problem dependent objective function that quantifies the optimality of an individual in a chromosome.

(ii) Selection

Parent chromosomes are selected from the problem space using some standard selection mechanisms like Roulette Wheel selection, Tournament selection, Rank based selection, truncate selection and Boltzmann selection. The result of all these selection mechanisms is to produce the best individual (chromosome) from the search space for the next iteration. The worst chromosomes are replaced by the best individuals in the next iteration which has the lowest probability.

(iii) Recombination

The genetic operators of GA include reproduction, crossover and mutation are commonly applied to problems of GA.

(iv) Reproduction

This operator is applied to an individual yields an offspring that is identical as the parent chromosome. There is no change in the genetic traits of the individual that is to be considered for the next generation.

(v) *Crossover*

This operator is also called as reproduction or recombination operator which is the primary operator which helps in generating the new offspring for the next generation. Generated offspring will not be identical with any of its parents. Every pair of individuals is not used in crossover. Generally single point and two point crossover is usually performed.

Crossover is done with two parent individuals to form a new offspring for the further generation. The new offspring is produced with the combination of the parental traits. For the parent chromosomes in a single point crossover parent 1 and parent 2 are given below.

- Parent 1 : 1111000011110000
- Parent 2 : 1100110011001100
- After crossover the resultant offspring will be
- Offspring 1 : 1111000**11001100**
- Offspring 2 : **11001100**11110000

In two point crossover, two cut points are chosen in both the parents and the offspring's are produced by exchanging the genetic materials as segments between the cut points. Suppose the crossover points randomly occur after the fourth and the thirteenth bit of the parent chromosome, then the offspring produced after the two point crossover are:

- Offspring 1 :1111**11001100**0000
- Offspring 2 :**1100**000011111**1100**

Similarly, Multipoint crossover is also used in which several crossover points are used for exchanging the genetic materials.

(vi) *Mutation*

New genetic traits are included into the existing individuals in the mutation operator. The genes value is randomly changed with in the chromosome. . Mutation is done to have diversity among chromosomes without changing the characteristics of the parent.

In the single point mutation, the gene is randomly chosen and it is mutated to produce the offspring.

- Parent : 1111000**0**
- Offspring : 1111000**1**

The last bit of the parent chromosome is mutated to generate the new offspring.

In the Multi point crossover any number of genes are randomly selected from the parent chromosome and mutated. The offspring produced after Multi point crossover is:

- Parent : 11**1**10**0**0**0**
- Offspring : 11**0**10**1**0**1**

After performing all the genetic operators in the dataset, the fitness value obtained after last iteration will be of optimum value. Obtained value is then sorted in the reverse order from which lower fitness value can calculated easily. The gene with lowest fitness value is calculated as an outlier.

IV.CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel algorithm for outlier detection using genetic algorithm. Genetic Algorithms (GAs) are adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. As such they represent an intelligent exploitation of a random search used to solve optimization problems. Although randomized, GAs are by no means random, instead they exploit historical information to direct the search into the region of better performance within the search space. It is better than conventional AI in that it is more robust. Unlike older AI systems, they do not break easily even if the inputs changed slightly, or in the presence of reasonable noise. Also, in searching a large state-space, multi-modal state-space, or n-dimensional surface, a genetic algorithm may offer significant benefits over more typical search of optimization techniques.

The future work is done by implementing the proposed work for more dataset of various types with necessary changes in the algorithm to make it more proficient. It has also been intended to apply the proposed system in distributed environment, to obtain better processing speed and performance of the algorithm.

REFERENCES

[1]. Ben-Gal I., Outlier detection, In: Maimon O. and Rockach L. (Eds.) “Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers," Kluwer Academic Publishers, 2005, ISBN 0-387-24435-2.

[2]. Kamlesh Kumar & Bidyut Kr. Patra," Outlier Detection for Large Datasets", ISSN (Print): 278-5140, Volume-2, Issue – 1, 2013.

[3]. P. Vishnu Raja,Dr. V. Murali Bhaskaran," An Effective Genetic Algorithm for Outlier Detectio n", International Journal of Computer Applications (0975 – 8887),Volume 38– No.6, January 2013.

[4]. Edwin M. Knorr,, Raymond T. Ng, Vladimir Tucakov," Distance-based outliers: algorithms and applications", Edited by. J. Widom. Received February 15, 1999 / Accepted August 1, 1999.

[5]. Knorr, E.M. and Ng, R. (1998). Algorithms for Mining Distance-Based Outliers in Large Datasets., Proceeding VLDB, pp.392-403.

IJETED