DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

GENETIC ALGORITHM BASED OUTLIER DETECTION USING INFORMATION CRITERION

by
Özlem GÜRÜNLÜ ALMA

June, 2009
İZMİR

GENETIC ALGORITHM BASED OUTLIER DETECTION USING INFORMATION CRITERION

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Statistics Program

by
Özlem GÜRÜNLÜ ALMA

June, 2009
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled "**GENETIC ALGORITHM BASED OUTLIER DETECTION USING INFORMATION CRITERION**" completed by **ÖZLEM GÜRÜNLÜ ALMA** under supervision of **PROF.DR. SERDAR KURT** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Serdar KURT

Supervisor

Assoc. Prof. Dr. Güçkan YAPAR	Assist. Prof. Dr. Aybars UĞUR
Thesis Committee Member	Thesis Committee Member

Prof.Dr. Hüseyin TATLIDİL	Assoc. Prof. Dr. Kaan YARALIOĞLU
Examining Committee Member	Examining Committee Member

Prof.Dr. Cahit HELVACI
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

First and foremost, I would like to express deeply felt thanks to my thesis advisor, Professor Serdar KURT, for helping me to successfully complete this dissertation. He is more than an advisor; he is a guide and his broad knowledge, interest, tenacity, enthusiasm, criticism, and constant encouragement have been essential in my formation as a researcher. Words are not enough to express my thanks to him for everything.

I am also most grateful to thank my co-advisor Assistant Prof. Dr. Aybars UĞUR whose energy, enthusiasm, insight and vast experience have been a source of inspiration. My appreciation goes to him for spending part of their time and making valuable suggestions to improve the quality of my work.

I would like to thank my dissertation committee member, Assoc. Prof. Dr. Güçkan YAPAR who made many valuable suggestions and gave constructive advice.

Special thanks are due to Professor Mustafa DİLEK, for all his supports, helpful suggestions, important advice and constant encouragement during my academic life. I also want to thank my friend Yalçın İŞLER for him helps and time spent to assist me in different stage of my thesis.

I would like to thank Assoc. Prof. Dr. C. Cengiz ÇELİKOĞLU, for kindness and all his supports since I began to work at Department of Statistics. Also, thank you all of department's staff for all their dedication and tremendous efforts to coordinate such wonderful working environment and for providing great services for our students.

I have been a very fortunate woman for sharing my life with Battal ALMA, my husband. Without his love, relentless support, understanding and continuous encouragement, it had not been possible to accomplish this goal. I would like to express my deepest gratitude, admiration and love for him. Last but by no means least, special thank also to my family for their support, love, and encouragement throughout my life.

Özlem GÜRÜNLÜ ALMA

GENETIC ALGORITHM BASED OUTLIER DETECTION USING INFORMATION CRITERION

ABSTRACT

Outlier, abnormal or unusual observation can be defined as an observation that lies outside the overall pattern of a distribution. Diagnostic methods for identifying a single outlier or influential observation in a linear regression model are relatively simple from both analytical and computational points of view. However, if the data set contains more than one outlier, which is likely to be the case in most data sets, the problem of identifying such observations becomes more difficult because of the masking and swamping effects.

In this thesis, Genetic Algorithm (GA) based outlier detection using information criteria in multiple regression models has been studied. A GA was allowed simultaneous detection of outliers in data sets. Thus, this method is to overcome the problems of masking and swamping effects. It is derived additional penalized value of information criteria for Akaike Information Criterion (AIC) and Information Complexity Criterion (ICOMP) and named as AIC' and ICOMP' respectively in this study. They have been used as the fitness function of genetic algorithms to detect outliers in multiple regression. The simulation study has been performed to compare consistency and robustness properties of AIC' and ICOMP' against corrected Bayesian Information Criterion (BIC'). Simulation results of AIC', BIC' and ICOMP' obtained from different number of sample sizes, different penalized Kappa values of information criterion and different number of explanatory variables for different percentage of outlier in dependent variables. The numerical example and simulation results clearly show a much improved performance of the proposed approach in comparison to existing method especially followed by applying the ICOMP' approach in order to accurately (robustly) detect the outliers.

Keywords: Genetic algorithms, Simultaneous outlier detection, Information criterion, AIC, BIC, and ICOMP Information criterion, Variable Selection, Multiple regression, Penalization.

BİLGİ KRİTERLERİ KULLANARAK GENETİK ALGORİTMA TABANLI AYKIRI DEĞER TESPİTİ

ÖZ

Aykırı değer, normal olmayan veya alışılmadık gözlem, bir dağılımın genel modeli dışında kalan gözlem olarak tanımlanabilir. Doğrusal regresyon modelinde, tek bir aykırı değeri veya etkili gözlemi belirleme yöntemleri analitik ve sayısal açıdan nispeten daha basittir. Bununla birlikte, birçok veri setinde karşılaşılan ve veri setinin birden fazla aykırı değer içermesi durumlarında, bu tür gözlemlerin belirlenmesi maskeleme ve batırma, sürükleme etkisinden dolayı oldukça güçleşmektedir.

Bu tezde, bilgi kriterleri kullanarak Genetik Algoritma (GA) tabanlı çoklu regresyon modellerinde aykırı değerlerin belirlenmesi çalışılmıştır. GA, veri kümelerinden eş zamanlı olarak aykırı değerlerin tespit edilmesini sağlar. Böylelikle, bu yöntem maskeleme ve batırma, sürükleme etkilerinin oluşturmuş olduğu sorunların üstesinden de gelmektedir. Çalışmada Akaike Bilgi Kriteri (AIC) ve Bilgi Karmaşıklığı Kriteri (ICOMP) için ek cezalandırma değeri türetilmiş ve bu bilgi kriterleri AIC' ve ICOMP' olarak adlandırılmıştır. Bu kriterler, çoklu regresyonda aykırı değerlerin tespiti için genetik algoritmanın uygunluk fonksiyonu olarak kullanılmıştır. AIC' ve ICOMP' bilgi kriterlerinin tutarlılık ve sağlamlılık özelliklerinin, tutarlı Bayes Bilgi Kriterine (BIC') karşı karşılaştırmak için benzetim çalışması gerçekleştirilmiştir. AIC', BIC' ve ICOMP''ın benzetim çalışması sonuçları, farklı sayıda örneklem büyüklükleri, farklı cezalandırma değeri, farklı sayıda açıklayıcı değişken ve bağımlı değişkenin farklı miktarda aykırı değer içermesi durumlarında elde edilmiştir. Çeşitli örnekler ve benzetim çalışması sonuçları açıkça göstermiştir ki önerilen yaklaşımlardan özellikle ICOMP' yaklaşımı aykırı değerleri doğru bir şekilde tespit etmektedir.

Anahtar sözcükler: Genetik algoritma, Eş zamanlı aykırı değer tespiti, Bilgi kriteri, AIC', BIC' ve ICOMP' bilgi kriteri, Değişken seçimi, Çoklu regresyon, Cezalandırma.

CONTENTS

Page

CHAPTER ONE
INTRODUCTION

1.1 Introduction

A genetic algorithm is a search technique used in computing to find true or approximate solutions to optimization and search problems. It is a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover.

For the last decade or so, the dimension of machine-readable data sets has increased impressively. Moreover, some processes such as on-line analytic processing allow rapid retrieval of data from data warehouse or huge databases. Presently, many of the advanced computational methods for extracting information from large quantities of data, or data mining methods, are developed, e.g., artificial neural networks, Bayesian networks, decision trees, genetic algorithms, and statistical pattern recognition. These developments have created a new range of challenges and opportunities for data analysis. However, there are potential quality problems with real data and databases which are generally contain amount of exceptional values or outliers.

Outliers are defined as the observations or records which appear to be inconsistent with the remainder group of the data. A well quoted definition of outliers is given by Hawkins (1980). This definition described an outlier as an observation that deviates so much from other observations as to arouse suspicion that is generated by a different mechanism. They may be generated by a different mechanism corresponding to normal data and may be due to sensor noise, process disturbances, instrument degradation, and/or human-related errors. It is futile to do data based analysis when data are contaminated with outliers because outliers can lead to model misspecification, biased parameter estimation and incorrect analysis results. The majority of outlier detection methods are based on an underlying assumption of identically and independently distributed data, where the location and the scatter are the two most important statistics for data analysis in the presence of outliers (Liu et al., 2004).

In the quest for data analysis, the issue of data quality has been found to be one of the important ones. It is commonly accepted that one of the most difficult and costly tasks in large-scale data analysis. Data quality process is trying to obtain clean and reliable data. Isolated outliers may also have positive impact on the results of data analysis. Many have estimated that as much as half to three fourths of a project's effort is typically spent on this part of process.

In many data analysis tasks a large number of variables are being recorded or sampled. One of the first steps towards obtaining a coherent analysis is the detection of outlying observations. An exact definition of an outlier often depends on hidden assumptions regarding the data structure and the applied detection method.

Although outliers are often considered as an error or noise, they may carry important information. Detected outliers are candidates for aberrant data that may otherwise adversely lead to model misspecification, biased parameter estimation and incorrect results. It is therefore important to identify them prior to modeling and analysis, especially if the data set contains more than one outlier, which is likely to be the case in most data sets, the problem of identifying such observations becomes more difficult because of the masking and swamping effects (Acuna & Rodriguez, 2004; Shekhar & Chawla, 2002).

Outlier detection methods have been suggested for numerous applications, such as credit card fraud detection, clinical trials, voting irregularity analysis, data cleansing, weather prediction, and other databases tasks. There are numerous methods for outlier detection in the literature. Barnett and Lewis (1994) give a lot of information about the outliers and their detection methods. Existing researches try to define algorithms to detect outliers based on distance or density. Some existing techniques for detecting outliers are clustering based methods, distance based methods, density based methods, subspace based methods, and statistical approaches (Aggarwal & Yu, 2001; Aggarwal & Yu, 2005; Agrawal et al., 2005; Breuning et al., 2000; Ester et al., 1998; Knorr & Ng, 1998). Among these techniques, statistical approaches and distance based outlier detection methods are the most popular in use.

The main aim of this study is to develop on outliers detecting method by using genetic algorithm. We propose that a simultaneous procedure for identification of outliers using new approaches of information criterion (AIC', BIC' and ICOMP') which can identify and test multiple outliers without suffering masking and swamping effects. The performance of these new information criterion approaches considered with generating experimental data. It is shown the behavior of new approaches for different sample sizes and different percentages of contaminated outliers by simulation on multiple regression models. That is, the outliers were produced by adding a given amount of percentages to each dependent variable. It is also studied on the effects of Kappa coefficients which are the penalized values of information criteria and obtained results for different values of them. Chapter 2 gives information about the Evolutionary and the Genetic Algorithm methods. Chapter 3 contains information about the outlier in databases and the outlier detection. Chapter 4 contains summary information about the information criteria such as Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and Information Complexity Criterion (ICOMP) which have been used as a fitness function of GA for detecting of outliers from multiple regression models. Chapter 5 contains information about the outlier detection methods using information criteria that are recently proposed and described our new approaches of information criteria. Details are given information on the performance of the method that we propose multiple outlier detection procedures for various configurations of data sets with outliers. Also the simulation results and conclusions are given in this chapter.

CHAPTER TWO
THE EVOLUTIONARY AND GENETIC ALGORITHMS

2.1 The Evolutionary Algorithm

Evolution is a method of searching among an enormous number of possibilities for solutions. In biology the enormous set of possibilities is the set of possible genetic sequences, and the desired solutions are organisms well able to survive and reproduce in their environments. Evolution can also be seen as a method for designing innovative solutions to complex problems. The fitness of a biological organism depends on many factors for example, how well it can weather physical characteristics of its environment and how well it can compete with or cooperate with the other organisms around it. The fitness criteria continually change as creatures evolve, so evolution is searching a constantly changing set of possibilities. Searching for solutions in the face of changing conditions is precisely what is required for adaptive computer programs. Furthermore, evolution is a massively parallel search method rather than work on one species at a time, evolution tests and changes millions of species in parallel (Mitchell, 1999).

The idea behind evolutionary algorithms comes from the biological method of evolution where selective pressures are applied to populations of organisms to evolve behaviors and features to allow for survival. Basically, if a difficult and complex search space exists with a solution somewhere inside it, that solution can be found by properly specifying the survival criterion of individuals and allowing for the evolutionary algorithm to search the space. The survival criterion is generally referred to as the fitness of a candidate solution. Since each individual in a genetic population represents a possible solution, that solution can be evaluated and given a fitness describing how well it solves the problem. Then, the fitness of each candidate solution in a population is used to drive the creation of a new population.

As with all algorithms, evolutionary algorithms take an input and return the desired output. However, in the case of evolutionary algorithms, it is not known how good the output will be. A desired performance is specified, but may be too complex for the

algorithm to find output and produces desired performance. Evolutionary algorithms can be represented by the general algorithm:

$$x[t+1] = s(v(x[t])) \tag{2.1}$$

Fogel (1998) described this algorithm: a population of candidate solutions to the problem at some time is denoted by x[t]. Random variations v, and selection methods s, are applied to the population at x[t] to produce a new population at the next time step, x[t+1] (Fogel,1998).

The general scheme of an Evolutionary Algorithm can be given as in Figure 2.1.

```
BEGIN
INITIALISE population with random candidate solutions;
EVALUATE each candidate;
REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
        1 SELECT parents;
        2 RECOMBINE pairs of parents;
        3 MUTATE the resulting offspring;
        4 EVALUATE new candidates;
        5 SELECT individuals for the next generation;
END
```

Figure 2.1 The evolutionary algorithm in pseudo-code

It is easy to see that this scheme falls in the category of generate and test algorithms. The evaluation function represents a heuristic estimation of solution quality and the search process is driven by the variation and the selection operators. Evolutionary Algorithms (EA) set a number of features that can help to position them within in the family of generate and test methods, these are:

- EA is population based, they process a whole collection of candidate solutions simultaneously,
- EA mostly uses recombination to mix information of more candidate solutions into a new one and,
- EA is stochastic.

There are the various dialects of evolutionary computing. For instance, the representation of a candidate solution is often used to characterize different streams. Typically, the candidates are represented by strings over a finite alphabet in Genetic Algorithms (GA), real-valued vectors in Evolution Strategies (ES), finite state machines in classical Evolutionary Programming (EP) and trees in Genetic Programming (GP). Technically, a given representation might be preferable over others if it matches the given problem better, that is, it makes the encoding of candidate solutions easier or more natural. For instance, for solving an optimization problem the straightforward choice is to use bit-strings of length n, where n is the number of logical variables, hence the appropriate EA would be a Genetic Algorithm. For evolving computer programs that can play checkers trees are well-suited, thus a GP approach is likely (Eiben & Smith, 2003).

2.2 The Genetic Algorithms (GAs)

It is very likely to be most widely known type of EA which is applied in science and engineering as stochastic search algorithms for solving optimization problems. The Figure 2.2 shows classes of search algorithms and where is the genetic algorithm in these search algorithms.

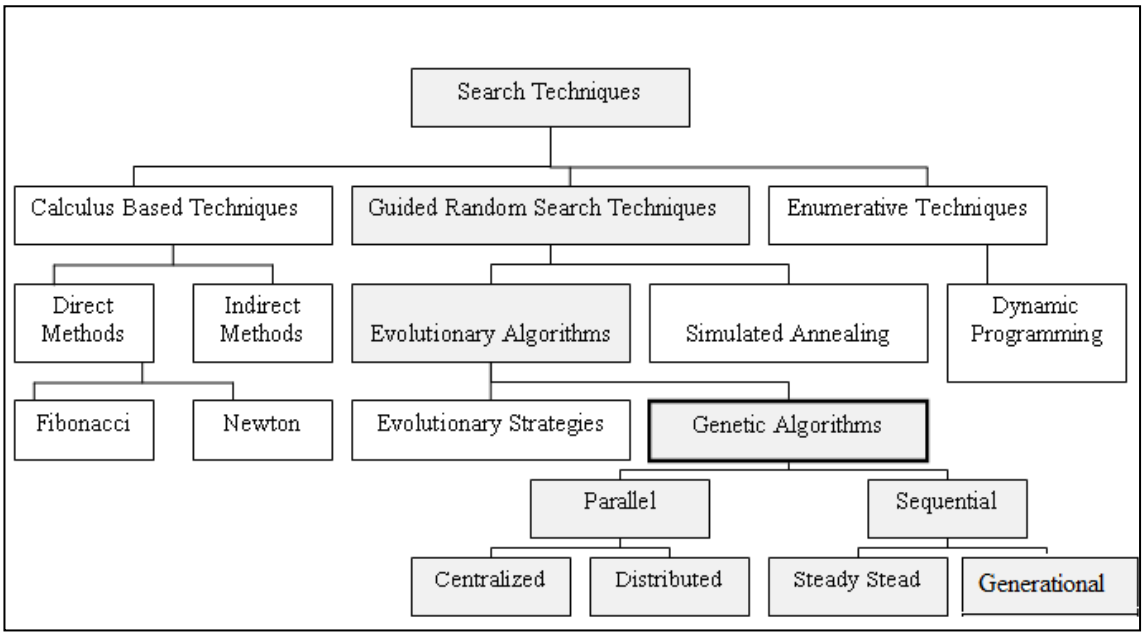

Figure 2.2 Classes of search algorithms (http://deron.csie.ncue.edu.tw/oop/GATutorial_deron.pdf)

In contrast to optimization techniques, GAs work with coding of parameters, rather than the parameters themselves. It is based on the genetic process of biological organisms. Over many generations, natural populations evolve according to the principle of natural selection and survival of the fittest.

GAs were first introduced by John H. Holland in his fundamental book Adaptation in Natural and Artificial Systems in 1975 (Bäck, 1996; Holland, 1975). Holland presented the algorithm as an abstraction of biological evolution and his schema theory laid a theoretical foundation for GAs. By mimicking this process, genetic algorithms are able to evolve solutions to real world problems, if they have been suitably encoded.

The power of GAs comes from the fact that the technique is robust, and can deal successfully with a wide range of problem areas, including those which are difficult for other methods to solve. GAs are not guaranteed to find the best solution to a problem, but they are generally good at finding global optimum solutions with a resonable amount of time and computational effort. The properties of GAs;

- The most important point is that GAs are parallel. Most other algorithms are serial and can only explore the solution space to a problem in one direction at a time, and, if the solution they discover turns out to be suboptimal, there is nothing to do but abandon all work previously completed and start over. However, since GAs have multiple offspring, they can explore the solution space in multiple directions at once. If one path turns out to be a dead end, they can easily eliminate it and continue work on more promising avenues, giving them a greater chance each run of finding the optimal solution (Goldberg, 1989; Mitchell, 1999).

- Another area in which GAs excel is their ability to manipulate many parameters simultaneously (Forrest, 1993). Many real world problems cannot be stated in terms of a single value to be minimized or maximized, but must be expressed in terms of multiple objectives, usually with tradeoffs involved: one can only be improved at the expense of another. GAs are very good at solving such

problems: in particular, their use of parallelism enables them to produce multiple equally good solutions to the same problem, possibly with one candidate solution optimizing one parameter to another candidate optimizing a different one, and a human overseer can then select one of these candidates to use (Haupt & Haupt, 1998).

- One of the qualities of GAs which might at first appear to be a liability turns out to be one of their strenghts: namely, GAs know nothing about the problems they are deployed to solve. Instead of using previously known domain specific information to guide each step and making changes with a specific eye towards improvement, as human designers do, they are blind watchmakers (Dawkins, 1996); they make random changes to their candidate solutions and then use the fitness function to determine whether those changes produce an improvement.

Although GAs has proven to be an efficient and powerful problem solving method, they have certain limitations. Some of them are as follows,

- The first and most important, consideration in creating a genetic algorithm is defining a representation for the problem. The language used to specify candidate solutions must be robust. There are two main ways of achieving this. The first, which is used by most genetic algorithms, is to define individuals as lists of numbers: binary valued, integer valued, or real valued, where each number represents some aspect of a candidate solution. In another method, genetic programming, the actual code does change. It represents individuals as executable trees of code.

- The problem of how to write the fitness function must be carefully considered so that higher fitness is attainable and actually does equate to a better solution for the given problem. If the fitness function is chosen poorly or defined imprecisely, the genetic algorithm may be unable to find a solution to the problem, or may end up solving the wrong problem. An example of this can be found in (Graham, 2002), in which researchers used an evolutionary algorithm

in conjunction with a reprogrammable hardware array, setting up the fitness function to reward the evolving circuit for outputting an oscillating signal.

- One type of the problem that GAs have difficulty dealing with are problems with deceptive fitness function, those where the locations of improved points give misleading information about where the global optimum is likely to be found (Mitchell, 1999).

- One well known problem that can occur with a GA is known as premature convergence. If an individual that is more fit than most of its competitors emerges early on in the course of the run, it may reproduce so abundantly that it drives down the population's diversity too soon, leading the algorithm to convergence on the local optimum that individual represents rather than searching the fitness landscape thoroughly enough to find the global optimum. (Forrest, 1993). This is an especially common problem in small populations where even chance variations in reproduction rate may cause one genotype to become dominant over others.

- Finally, Forrest (1993), Haupt and Haupt (1998) advise against using GAs on analytically solvable problems. It is not that GAs cannot find good solutions to such problems; it is merely that traditional analytic methods take much less time and computational effort than GAs and, unlike GAs, are usually mathematically guaranteed to deliver the one exact solution. Of course, since there is no such thing as a mathematically perfect solution to any problem of biological adaptation, this issue does not arise in nature.

This chapter is organized as follows. Firstly, it is given biological terminology for a better understanding of GA. Then, the next and other subsections provide details of individual's steps of a typical genetic algorithm and introduce several popular genetic operators. Also, subsections give a brief overview of designing principled efficiency-enhancement techniques to speed up genetic algorithms.

2.2.1 Biological Terminology and Explanation of Genetic Algorithms

Each cell of a living creature consists of a certain set of chromosomes which are made of genes. Each gene encodes one or more characters that can be passed on to the next generation. Each gene can be in different states, called alleles. Genes are located at certain positions on the chromosome. The cell many creatures has more than one chromosome. The entire set of chromosomes in the cell is called the genome. In the natural reproduction process, pieces of gene material are exchanged between the two parents' chromosomes to form new genes. This process is called recombination or crossover. Genes in the offspring are subject to mutation, in which a certain block of DNA in the gene undergoes a random change (Michalewicz, 1996; Mitchell, 1999).

In the GAs, a chromosome is used to represent a potential solution to a problem. Each solution has a representation made up of numbers of genes in GA. The various combinations of these genes of different alleles can produce different structures (genotypes) all of which can be seen as a different solution. Table 2.1 presents the explanations of the terms used in GA (Gen & Cheng, 2000).

Table 2.1 Explanation of genetic algorithm terms

Genetic Algorithms	**Explanation**
Chromosome (string, inidividual)	Solution (Coding)
Genes (bits)	Part of the solution
Locus	Position of gene
Alleles	Value of gene
Genotype	Encoded solution
Phenotype	Decoded solution

To understand the substructure of GA is important and the concept is explained as follows:

- Each individual in the population is called a chromosome which is denoted a string of symbols in GA, for example it is used a binary string form in Figure 2.3.

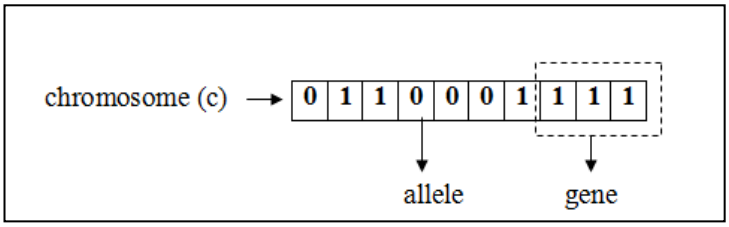

Figure 2.3 Structure of a chromosome

The chromosomes evolve through successive iterations, and the fitness of chromosomes is evaluated using fitness function. Fitter chromosomes have higher probabilities of being selected. After several generations, the algorithms converge to the best chromosome, which hopefully represents the optimal or suboptimal solution to the problem. For example, in a problem such as the traveling salesman problem, a chromosome represents a route, and a gene may represent a city (Goldberg, 1989).

- The gene is binary encoding of a single parameter.
- Alleles are denoted to be a gene and these are value of a gene. In biology, alleles are one of the functional forms of a gene.
- The genotype is the genetic composition of an organism. The information contained in the chromosomes, and
- The phenotype is the environmentally and genetically determined traits of an organism. These traits are actually observed at phenotype while not observed at genotype. Genetic operators work on the level of the genotype, whereas the evaluation of the individuals is performed on the level of the phenotype.

2.2.2 General Structure of Genetic Algorithm

GA is stochastic search techniques based on the mechanism of natural selection and natural genetics. It imitates basic principles of life and applies genetic operators like mutation, crossover, or selection to a series of alleles which is the equivalent of a chromosome in nature (Holland, 1975).

A GA operates on a population of individuals or chromosomes representing potential solutions to a given problem. Each chromosome is assigned a fitness value

according to the result of the fitness function. The selection mechanism favors individuals of better fitness function value to reproduce more often than worse ones when a new population is formed. Recombination allows for the mixing of parental information when this is passed to their descendants, and mutation introduces innovation in the population. Usually, the initial population is randomly initialized and evolution process is stopped after a predefined number of iterations (Azzaro-Pantel et. al., 1998). The general structure of GA is shown as Figure 2.4 (Grupe & Jooste, 2004).

1. **[Initialize]** The initial population of n chromosomes is generated randomly across the search space.
2. **[Evaluate]** Evaluate the fitness f(c) of each chromosome c in the population.
3. **[Offspring]** Create a new population by executing the following steps.
    a. **[Selection]** Select n parent chromosomes from the population according to their fitness.
    b. **[Crossover]** Recombine the parents with a certain crossover probability to form new offspring.
    c. **[Mutation]** Mutate the new offspring with certain mutation probability at each locus (position in chromosome).
4. **[Replace]** Replace the current population with the newly generated population.
5. **[Test]** If the termination conditions is satisfied, stop, and return the best chromosome found; otherwise go to step 2.

Figure 2.4 The General structure of genetic algorithm (Grupe & Jooste, 2004)

This process can be iterated until a candidate with sufficient solution is found or a previously set computational limit is reached. In this process there are two fundamental forces that form the basis of evolutionary systems (Eiben & Smith, 2003);

- Variation operators create the necessary diversity and,
- Selection acts as a force pushing quality.

The combined application of variation and selection generally leads to improving fitness values in sequence populations. It is easy to see such a process as if the evolution is optimizing, or at least approximating, by approaching optimal values closer and closer over its course. Evolution is often seen as a process of adaptation.

From this perspective, the fitness is not seen as an objective function to be optimized, but as an expression of environmental requirements. Matching these requirements more closely implies an increased viability, reflected in a higher number of offspring. The evolutionary process makes the population adapt to the environment better and better.

During selection fitter individuals have a higher chance to be selected than less fit ones, but typically even the weak individuals have a chance to become a parent or to survive. For recombination of individuals the choice of which pieces will be recombined is random. Similarly for mutation, the pieces that will be mutated within a candidate solution, and the new pieces replacing them, are chosen randomly.

The GA begins, like any other optimization algorithm by defining the optimization variables, the fitness function and fitness value. It ends by testing for convergence. A path through the components of GAs is shown as a flowchart in Figure 2.5 (Lee & El-Sharkawi, 2008).

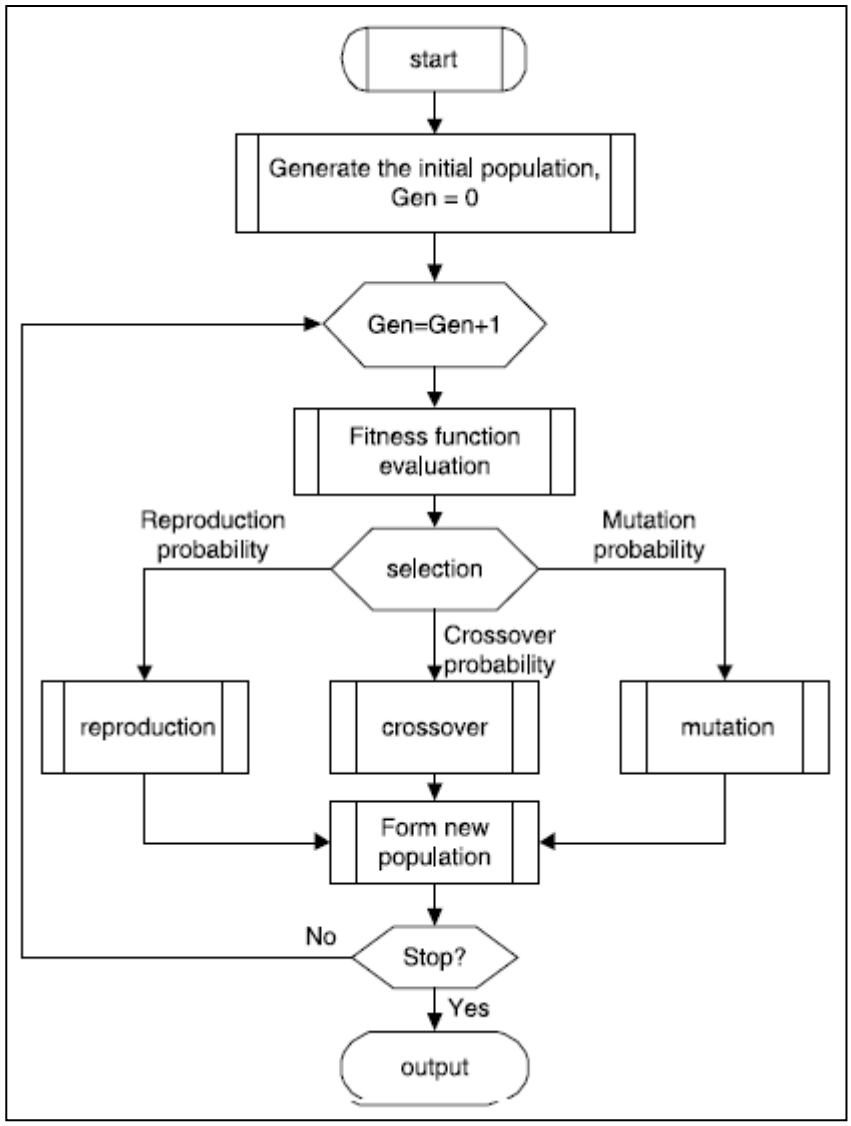

Figure 2.5 Flowchart of genetic algorithms (Lee & El-Sharkawi, 2008)

The major questions to consider are firstly the size of population, and secondly the method by which the individuals are chosen. The choice of the population size has been approached from several theoretical points of view, although the underlying idea is always of trade-off between efficiency and effectiveness. Intuitively, it would seem that there should be some optimal value for a given string length, on the grounds that too small a population would not allow sufficient room for exploring the search space effectively, while too large a population would so impair the efficiency of the method that no solution could be expected in a reasonable amount of computation.

The first stage of building genetic algorithm is to decide on the representation of a candidate solution to the problem. Without representations, no use of GA is possible; therefore, some representations are explained as follows.

2.2.3 Representation of Individuals or Encoding

The fitness function measures the fitness of an individual to survive, mate, and produce offspring in a population of individuals or chromosomes for a given problem. The GA will seek to maximize the fitness function by selecting the individuals. Therefore, chromosome representation is a very critical issue in the success of the GA for this reason. An appropriate representation must be capable of representing any possible solution for the problem and at the same the representation scheme must not support to include the infeasible solutions in the population if it is possible to do so. In contrast to traditional optimization techniques, GAs work with coding of parameters, rather than the parameters themselves. It is important to choose the right representation for the problem being solved. Getting the representation right is one of the most difficult parts of designing a good evolutionary algorithm. Often this only comes with practice and a good knowledge of the application domain.

Rothlauf (2006) defined the genotypic search space as φ_g which is either discrete or continuous, and the function $f(x):\varphi_g \rightarrow R$ assigns an element in R to every element in the genotype space φ_g. The optimization problem is defined by finding the optimal solution $\hat{x} = \max_{x \in \varphi_g} f(x)$, where x is a vector, and $\hat{x}$ is the global maximum. When using a representation it is had to introduce phenotypes and genotypes. Thus, the fitness function f can be decomposed into two parts. The first maps the genotypic space φ_g to phenotypic space φ_p, and the second maps phenotypic space to the fitness space R. Using the phenotypic space φ_p, it is obtained:

$$\begin{aligned} f_g(x_g):\varphi_g &\rightarrow \varphi_p \\ f_p(x_p):\varphi_p &\rightarrow R \end{aligned} \qquad (2.2)$$

where $f = f_p \circ f_g = f_p(f_g(x_g))$ is described by Rothlauf (2006). The genotype-phenotype mapping f_g is the used representation. f_p represents the fitness function and assigns a fitness value $f_p(x_p)$ to every individual $x_p \in \varphi_p$. The genetic operators are applied to the individuals in φ_g that means on the level of genotypes.

It is important note that the recombination and mutation operators working on candidates must match the given representation. In the following section it is looked more closely at some commonly used representations and the genetic operators that might be applied to them. It is important to stress, however that while the representations described here are commonly used. Although it is presented the representations and their associate operators separately, it frequently turns out in practice that using mixed representations is a more natural and suitable way of describing and manipulating a solution than trying to different aspects of a problem into a common form.

I. Binary Representations: This is the one of the earliest and simplest representations. Each gene is coded the bit string chromosome. The bit string length depends on the required numerical precision. The genotype consists simply of a string of binary digits a bit string (Eiben & Smith, 2003). When using the binary encoding, the search space is denoted by $\phi_g = \{0,1\}^{\ell}$,where ℓ is the length of a binary vector $x^g = \{x_1^g, x_2^g, ..., x_\ell^g\} \in \{0,1\}^{\ell}$ and $x^p \in [0,1]$ (Goldberg,1989).Each integer phenotype $x^p \in \phi_p = \{1,2,..., x_{max}\}$ is represented by a binary genotype x^g of length $\ell = [\log_2(x_{max})]$. The genotype-phenotype mapping f_g is defined as;

$$x^p = f_g(x^g) = \sum_{i=0}^{i-1} 2^i x_i^g \tag{2.3}$$

with x_i^g denoting the i^{th} bit of x^g (Rothlauf, 2006). An example of genotype-phenotype mapping is illustrated in the Figure 2.6.

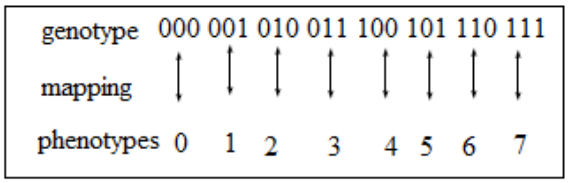

Figure 2.6 Genotype - phenotype mapping

For a particular application it must decide how long the string should be, and how it will interpret to produce a phenotype. In choosing the genotype-phenotype mapping for a specific problem, one has to make sure that the encoding allows that all possible bit strings denote a valid solution to the given problem and that, vice versa, all possible solutions can be represented.

One of the problems of coding numbers in binary is that different bits have different significance. This can be helped by using Gray coding, which is a variation on the way that integers are mapped on bit strings. The standard method has the disadvantage that the Hamming distance between two consecutive integers is often not equal to one (Eiben & Smith, 2003). For some problems, particularly those concerning Boolean decision variables, the genotype-phenotype mapping is natural, but frequently bit strings are used to encode other non-binary information. For example, we might interpret a bit-string of length 80 as ten 8-bit integers, or five 16-bit real numbers. Usually this is a mistake, and better results can be obtained by using the integer or real valued representations directly.

II. Integer Representations: Binary representations are not always the most suitable if the problem more naturally maps onto a representation where different genes can take one of a set of values. One obvious example of when this might occur is the problem of finding the optimal values for a set of variables that all take integer values. These values might be unrestricted, or might be restricted to a finite set: In either case an integer encoding is probably more suitable than binary encoding. When designing the encoding and variation operators, it is worth considering whether there are any natural relations between the possible values that an attribute can take (Eiben & Smith,

2003). If this representation is used, there are x^{ℓ} different individual possibilities and the size of search space increases from $|\varphi_g| = 2^{\ell}$ to $|\varphi_g| = x^{\ell}$.

III. Real-Valued or Floating-Point Representations: Often the most sensible way to represent a candidate solution to a problem is to have a string of real values. This occurs when the values that it is wanted to represent as genes come from a continuous rather than a discrete distribution. Of course, on a computer the precision of these real values is actually limited by the implementation so we will refer to them as floating-point numbers. When using real valued representations, the search space φ_g is defined as $\varphi_g = R^{\ell}$ where ℓ is the length of the real valued chromosome.

IV. Permutation Representations: Many problems naturally take the form of deciding on the order in which a sequence of events should occur. The most natural representation of such problems is as a permutation of a set of integers. One immediate consequence is that while an ordinary GA string allows numbers to occur more than once, such sequences of integers will not represent valid permutations. It is clear that we need new variation operators to preserve the permutation property that each possible allele value occurs exactly once in the solution. This representation is used as firstly i^{th} element of the representation denotes the event that happens in that place in the sequence. In the second, the value of the i^{th} element denotes the position in the sequence in which the i^{th} element happens. For example; for the four cities [A, B, C, D], and the permutation [3, 1, 2, 4], the first encoding denotes the tour [C, A, B, D] and the second [B, C, A, D] (Eiben & Smith, 2003).

2.2.4 Initial Population Generation

The initial population for GA is the first group of solutions among which the search begins. As declared in Reeves and Rowe (2003), the point in generating the initial population is that "every point in the search space or in other words any solution to the original problem could be reached from the solutions in the initial population by crossover only" and this could only be satisfied by the existence of each possible value

for each gene in the initial population. This emphasizes the importance on the way the initial population is generated.

The most common way of generating the initial population is doing this randomly without any control on the existence of alleles for genes. While this approach is in accordance with the stochastic nature of the GAs, individuals generated in this way do not necessarily cover the solution space.

Population size on the other hand, usually depends on the nature of the problem and, it is usually a user specified parameter, is one of the important factors affecting the scalability and performance of genetic algorithms. Reeves and Rowe (2003) denotes that the underlying idea of the population size is trade off between efficiency and effectiveness. As the population size decreases, the chances for exploring the search space effectively also decrease. Small population sizes might lead to premature convergence and yield substandard solutions However if the population size is too large, the efficiency of the application decreases due to the increased computation time. On the other hand, large population sizes lead to unnecessary expenditure of valuable computational time (Sastry et al., 2005).

2.2.5 Fitness Function

In nature the fitness relates to the ability of the organism to survive and, reproduce, that is, organisms with a better fitness score are more likely to be selected for reproduction, in genetic algorithms the fitness is the evaluated result of a user defined objective function (Mitchell, 1999). Each chromosome is evaluated and assigned a fitness value after the creation of an initial population. On the basis of this value, the selection process decides which of the genomes are chosen for reproduction.

The fitness function is a black box for the GA. Internally; this may be achieved by a mathematical function, a simulation model, or a human expert that decides the quality of a chromosome. At the beginning of the iterative search, the fitness function values for the population members are usually randomly distributed and wide spread

over the problem domain. As the search evolves, particular values for each gene begin to dominate. The fitness variance decreases as the population converges. This variation in fitness range during the evolutionary process often leads to the problem of premature convergence and slow finishing.

Premature convergence occurs when the genes from a few comparatively fit individuals may rapidly come to dominate the population, causing it to converge on a local maximum. To overcome this problem, the way individuals are selected for reproduction must be modified. One needs to control the number of reproductive opportunities each individual gets so that it is neither too large nor too small.

Slow finishing is the converse problem to premature convergence. After many generations, the population will have largely converged, but may still not have precisely located the global maximum. The average fitness will be high, and there may be little difference between the best and average individuals. As with premature convergence, fitness scaling can be prone to over compression due to just one super poor individual (Beasley et al., 1993).

2.2.6 Parent Selection Methods

Selection is a process in which chromosomes are copied according to their fitness function value. It is used for two objectives; for determining the mates to reproduce and for determining the fitter chromosomes which will be maintained in the next generation. This method has a magnificent effect on results. If the selector picks only the best individual, then the population will quickly converge to that best value. The selector should also pick individuals that are not so good, but have good genotype to avoid from early convergence. For a detailed explanation of a variety of techniques, Haupt and Haupt (1998), Reeves and Rowe (2003), Beasley et al. (1993), Goldberg and Deb (1991) could be referred to. There are several parent selection techniques. Some of these could be summarized as follows.

I. Fitness Proportional Selection (FPS): After the fitness values for the chromosomes are calculated, selection probabilities related to each chromosome is calculated regarding the total fitness of the population. FPS includes methods such as roulette wheel selection (Goldberg, 1989; Holland, 1975) and stochastic universal selection (Baker, 1985; Grefensette & Baker, 1989). In roulette wheel selection, each individual in the population is assigned a roulette wheel slot sized in proportion to its fitness. That is, in the biased roulette wheel, good solutions have a larger slot size than less fit solutions. The roulette wheel is spun N (population size) to obtain a reproduction candidate. The roulette wheel selection procedures are detailed as follows in Figure 2.7 (Sastry et al., 2005):

1. Evaluate the fitness, f_i, of each individual in the population.
2. Compute the probability p_i, of selecting each member of the population $p_{i=} \frac{f_i}{\sum_{j=1}^{n} f_j}$, where n is the population size.
3. Calculate the cumulative probability q_i, for each individual $q_{i=} \sum_{j=1}^{i} p_j$
4. Generate a uniform random number $r \in (0,1]$.
5. If $r < q_1$ then select the first chromosome, c_1, else select the chromosome c_i.
6. Repeat steps 4 and 5 n times to crate n candidates in mating pool.

Figure 2.7 The Roulette wheel selection procedures (Sastry et al., 2005)

For each choice, the probability that an individual f_i is selected for mating is $\frac{f_i}{\sum_{j=1}^{n} f_j}$ that is to say that the selection probability depends on the absolute fitness value of the individual compared to the absolute fitness values of the rest of the population. To illustrate, consider a population with four chromosome, n=4, with the fitness as shown in the below. The total fitness, $\sum_{j=1}^{n} f_j$ =25 + 15+ 12 +18= 70. The probability of selecting an individual and the corresponding cumulative probabilities are also shown in below.

Chromosome Number	1	2	3	4
Fitness, f_i	25	15	12	18
Probability, p_i	25/70=0.35	0.22	0.18	0.25
Cumulative probability, q_i	0.35	0.57	0.75	1

It is assume that a random number r is 0.64, then the third chromosome is selected as $q_2=0.57< 0.64 \leq q_3=0.75$.

There are some problems with this selection mechanism for instance, when fitness values are all very close together, there is almost no selection pressure, since the parts of the roulette wheel assigned to the individuals are more or less the same size, so selection is almost uniformly random and having a slightly better fitness is not very useful to an individual. Therefore, later in a run when some convergence has taken place and the worst individuals are gone, the performance only increases very slowly.

II. Ranking Selection: Rank based selection is another method that was inspired by the observed drawbacks of fitness proportionate selection. It preserves a constant selection pressure by sorting the population on the basis of fitness and then allocating selection probabilities to individuals according to their rank, rather than according to their actual fitness values. The mapping from rank number to selection probability is arbitrary and can be done in many ways, for example, linearly or exponentially decreasing, of course with the condition that the sum over the population of the probabilities must be unity (Eiben & Smith, 2003).

The usual formula for calculating the selection probability for linear ranking schemes is parameterized by a value s ($1< s\leq 2$). In the case of a generational GA, where μ is the total number of ranks. If this individual has rank μ, and the worst has rank 1, then the selection probability for an individual of rank i is (Eiben & Smith, 2003):

$$P_{lin-rank(i)} = \frac{(2-s)}{\mu} + \frac{2i(s-1)}{\mu(\mu-1)} \qquad (2.4)$$

An example of how the selection probabilities differ for a population of three different individuals with fitness proportionate and rank-based selection with different values is showed that in below. FP is fitness proportionate and LR is linear ranking selection (Eiben & Smith, 2003).

	Fitness	Rank	P_{setFP}	P_{setLR} (s=2)	P_{setLR} (s=1,5)
A	1	1	0.1	0	0.167
B	5	3	0.5	0.67	0.5
C	4	2	0.4	0.33	0.33
Sum	10		1	1	1

When a linear mapping is used from rank to selection probabilities the amount of selection pressure that can be applied is limited. This arises from the assumption that, on average, an individual of median fitness should have one chance to be reproduced, which in turn imposes a maximum value of s=2. If a higher selection pressure is required i.e., more emphasis on selecting individuals of above average fitness an exponential ranking scheme is often used of the form (Eiben & Smith, 2003):

$$P_{exp-rank(i)} = \frac{1 - e^{i}}{c} \tag{2.5}$$

The normalization factor c is chosen so that the sum of the probabilities is unity, i.e., it is a function of population size.

III. Tournament Selection: The previous two selection methods and the algorithms used to sample from their probability distributions relied on knowledge of the entire population. In certain situations, for example, if the population size is very large or if the population is distributed in some way obtaining this knowledge is either highly time consuming or at worst impossible. In yet other cases there might not be universal fitness definition at all (Eiben & Smith, 2003).

Tournament selection is an operator with the useful property that it does not require any global knowledge of the population. Instead it only relies on an ordering relation that can rank any two individuals. It is therefore conceptually simple and fast to implement and apply. The application of tournament selection to select μ parents work according to the procedure is showed in Figure 2.8 (Eiben & Smith, 2003).

```
BEGIN
    Set current_member=i=1
    While (current_member ≤ μ) do
       Pick k individuals  randomly,with or without replacement;
       Select the best of these k comparing  their fitness values;
       Denote this individual as i ;
       set mating pool [current_member]=[i];
       set current_member= current_member+1
    od
END
```

Figure 2.8 Tournament selection algorithm (Eiben & Smith, 2003)

The probability that an individual will be selected as the result of a tournament depends on four factors, namely:

- Its rank in the population. Effectively this is estimated without the need for sorting the whole population.

- The tournament size k. The larger the tournament, the more chance that it will contain members.

- The probability p that the most fit member of the tournament is selected. Usually this is 1 for deterministic tournaments, but stochastic versions are also used with p< 1. Clearly in this case there is lower selection pressure.

- Whether individuals are chosen with or without replacement. In the second case with deterministic tournaments, the k-1 least-fit members of the population can never be selected, whereas if the tournament candidates are picked with

replacement, it is always possible for even the least-fit member of the population to be selected.

2.2.7 Crossover Operators

After parents have been selected through one of the methods introduced above, they are randomly paired. The genetic operators are applied on these paired parents to produce offspring and it is the most important operator in a genetic algorithm. The purpose of crossover is to vary the individual quality by combining the desired characteristics from two parents (Booker et. al., 1997; Spears, 1997).

In most crossover operators, two parents are randomly selected and recombined with a crossover probability p_c which determines the chance that a chosen pair of parents undergoes this operator (Eiben & Smith, 2003). That is, a uniform random number r, is generated and if $r \leq p_c$, the two randomly selected individuals undergo recombination. Otherwise, that is if $r > p_c$, the two offspring are simply copies of their parents. The value of p_c can either be set experimentally, or can be set based on schema theorem principles (Goldberg, 1989; Godlberg, 2002; Sastry et. al., 2005).

The net effect is that in general the resulting set of offspring consists of some copies of the parents, and other individuals that represent previously unseen solutions. Over the years, numerous variants of crossover have been developed in the GA literature, and comparisons also have been made among these methods (Eshelman et. al., 1989). However, most of these studies rely on a small set of test problems, and thus it is hard to draw a general conclusion on which method is better than others.

A number of commonly used crossover techniques are explained as follows:

I. Crossover Operators for Binary Representations: There are three types of crossover techniques for binary representations of chromosomes. These are defined as below.

• **One-Point Crossover:** This is the traditional and the simplest way of crossover: a position is randomly chosen as the crossover point and then the two parts of the parents after the selected point are swapped to make two offspring. The Figure 2.9 shows this operation.

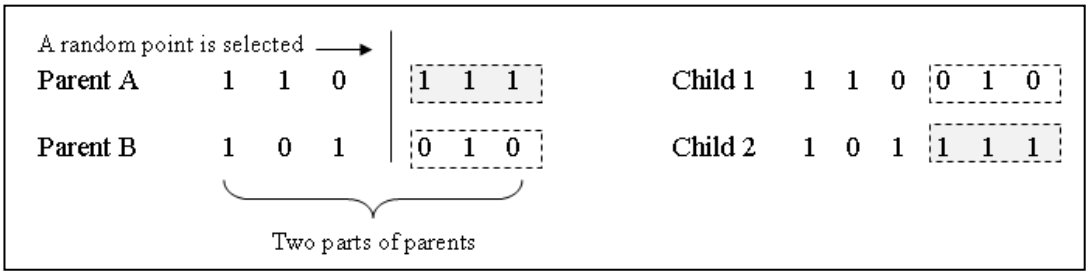

Figure 2.9 One-point crossover

• **N-Point Crossover:** One-point crossover can easily be generalized to n-point crossover, where the representation is broken into more than two segments of contiguous genes, and then offsprings are created by taking alternative segments from the two parents. In practice this means choosing n random crossover points in [0, k-1], k is crossover point, which is illustrated in below Figure 2.10 for n=2.

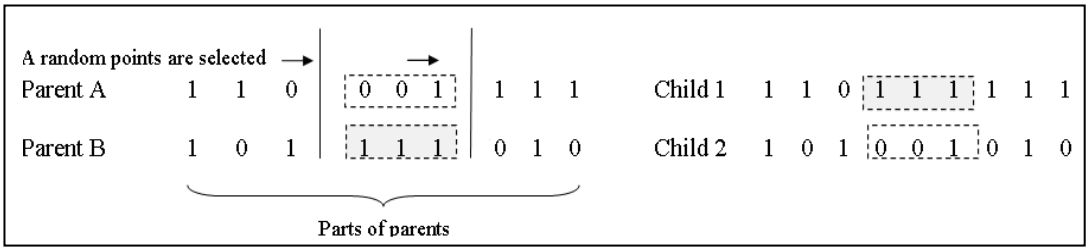

Figure 2.10 N-point crossover

• **Uniform Crossover:** The previous two operators work by dividing the parents into a number of sections of contiguous genes and reassembling them to produce offspring. In contrast to this, uniform crossover works by certain probability p_e, known as the swapping probability. Usually the swapping probability value is taken to be 0.5. In each position, if the value is below a parameter the gene is inherited from the first parent; otherwise from the second. The second offspring is created the inverse mapping. For example the array [0.35, 0.62, 0.18, 0.42, 0.83, 0.76, 0.39, 0.51, 0.36] of random variables drawn uniformly from [0,1] was used to decide inheritance and the offsprings are shown in the Figure 2.11 (Eiben & Smith, 2003; Spears, 1994; Syswerda, 1989).

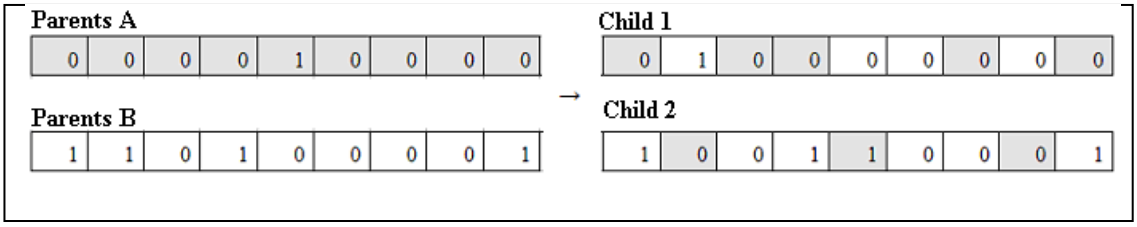

Figure 2.11 Uniform crossover (Eiben & Smith, 2003)

II. Crossover Operator for Integer Representation: For each gene has a higher number of possible allele values such as integers it is normal to use the same set of operators as for binary representations.

III. Crossover Operators for Floating-Point Representation: There are two options for recombining two floating-point strings:

- An allele is one floating-point value instead of one bit. This has the disadvantage that only mutation can insert new values into the population since recombination only gives us new combinations of existing floats. Recombination operators of this type for floating-point representations are known as discrete recombination and have the property that if an offspring z is creating from parents x and y then the allele value for gene i is given by $z_i \rightarrow x_i$ or y_i with equal likelihood (Haupt & Haupt, 1998).

- Using an operator that in each gene position creates a new allele value in the offspring that lies between those of parents x_i, y_i. Using the terminology (2.6),

$$\text{Child} = \alpha x_i + (1 - \alpha) y_i \qquad (2.6)$$

for some α in [0,1]. In this way recombination is now able to create new gene material, but it has the disadvantage that as result of the averaging process the range of the allele values in the population for each gene is reduced. Operators of this type are known as arithmetic recombination. This is the most commonly used operator and works by taking the weighted sum of the two parental alleles for each

gene (Wright, 1991). For example is chosen below terminology for crossover operator (Eiben & Smith, 2003),

$$\text{Child1} = \alpha\bar{x} + (1-\alpha)\bar{y} \qquad \text{Child2} = \alpha\bar{y} + (1-\alpha)\bar{x} \tag{2.7}$$

According to this terminology and for α=1/2 two offsprings will be identical for this operator and their values are as follows.

Parent $\bar{x}$										Child 1								
0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9		0.2	0.2	0.3	0.3	0.4	0.4	0.5	0.5	0.6
Parent $\bar{y}$									→	Child 2								
0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3		0.2	0.2	0.3	0.3	0.4	0.4	0.5	0.5	0.6

Figure 2.12 Crossover operators for floating-point representation (Eiben & Smith, 2003)

IV. Crossover Operators for Permutation Representation: A number of specialized crossover operators have been designed for permutations, which aim at transmitting as much as possible of the information contained in the parents especially that held in common. There are several operators for permutation problems of which the best known is order crossover. Order Crossover operator was designed by Davis (1991) for order based permutation problem. It recombined parents as (Eiben & Smith, 2003);

1. Choose two crossover points at random and copy the segment between them from the first parent (P1) into the first offspring.

2. Starting from the second crossover point in the second parent, copy the remaining unused numbers into the first child in the order that they appear in the second parent, wrapping around at the end of the list.

3. Create the second offspring in an analogous manner with the parent roles reversed.

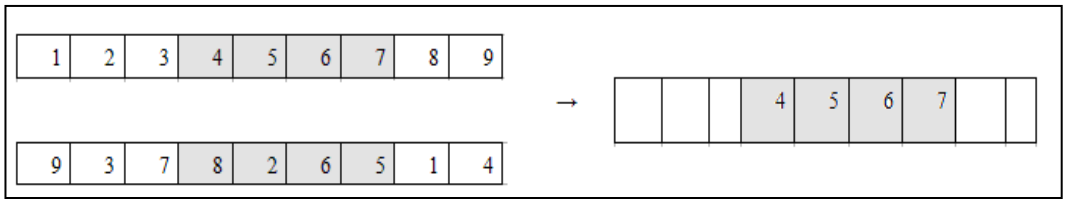

Figure 2.13 Crossover operators for permutation representation (Eiben & Smith, 2003)

Figure 2.13 is illustrated: step 1; copy randomly selected segment from first parent into offspring. Step 2 is illustrated below, copy rest of alleles in order they appear in second parent treating string as circle.

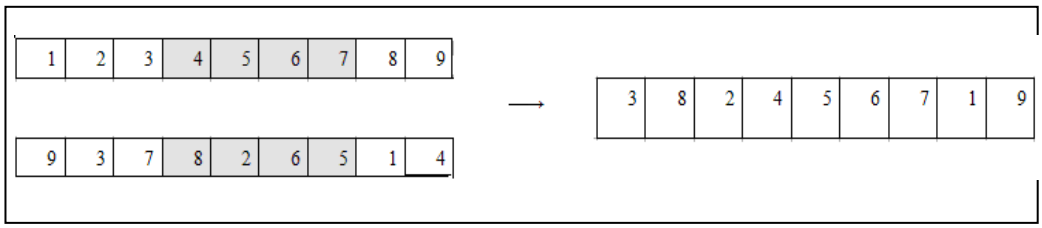

Figure 2.14 Crossover Operators for Permutation Representation (Eiben & Smith, 2003)

2.2.8 Mutation Operators

Mutation is a background operator which produces instinctive random changes in various chromosomes. A simple way to achieve mutation would be alter one or more genes. In GA, mutation serves the crucial role of either: replacing the genes lost from the population during the selection process so that they can be tried in a new context or, providing the genes that were not presented in the initial population.

Mutation use only one parent and create one child by applying some kind of randomized changed to the genotype. The form taken depends on the choice of encoding used, as does the meaning of the associated parameter, which is often referred to as the mutation rate. The mutation rate is defined as the percentage of the total number of genes in the population. The mutation rate controls the rate at which new genes are introduces into the population for trial. If it is too low, many genes that would have been useful are never tried out. If it is too high, there will be much random disorder, the offspring will start losing their resemblance to the parents, and the algorithm will lose the ability to learn from the history of the search.

I. Mutation Operator for Binary Representations: The most common mutation operator used for binary encoding considers each gene separately and allows each bit to flip (i.e. from 1 to 0 or 0 to 1) with a small probability $p_{m.}$ The actual number of values changed is thus not fixed, but depends on the sequence of random numbers drawn, so for an encoding of length L on average ($L*p_m$) value will be changed. A number of studies and suggestions have been made for the choice of suitable values for the bitwise mutation rate and it is worth noting at the outset that the most suitable choice to use depends on the desired outcome. For example does the application require a population in which all members have high fitness, or simply that one highly fit individual is found? However, most binary coded GAs use mutation rates in a range such that on average between one gene per generation and one gene per offspring is mutated.

II. Mutation Operator for Integer Representations: For integer encodings there are two principal forms of mutation used both of which mutate each gene independently with user-defined probability $p_{m.}$

Random Resetting Mutation: At this juncture the bit-string mutation of binary encodings is extended to random resetting, so that with probability p_m a new value is chosen at random from the set of allowed values in each position. This is the most suitable operator to use when the genes encode for cardinal attributes, since all other gene values are equally likely to be chosen (Eiben & Smith, 2003).

Creep Mutation: This mutation was designed for ordinal attributes and works by adding small value to each gene with probability p. Generally these values are sampled randomly for each position from a distribution that is symmetric about zero and is more likely to generate small changes than large ones. It should be noted that creep mutation requires a number of parameters controlling the distribution from which the random numbers are drawn and hence the size of the steps that mutation takes in the search space. Finding appropriate settings for these parameters may not be easy and it is sometimes common to use more than one mutation operator in joining from integer-based problems (Eiben & Smith, 2003).

III. Mutation for Floating-Point Representations: For floating point representations, the allele values as coming from a continuous rather than a discrete distribution so the forms of mutation described above no longer applicable. Instead, it is common to change the allele value of each gene randomly within its domain given by a lower L_i and upper U_i bound resulting in the following transformation:

$$\{x_1, x_1, ...x_n\} \rightarrow \{x_1', x_2', ..., x_i'\} \qquad \text{where} \qquad x_i' \subset \{L_i, U_i\}$$

Two types can be distinguished according to the probability distribution from which the new gene values are drawn: uniform and non-uniform mutation (Eiben & Smith, 2003).

Uniform Mutation: For this operator the values of x_i' are drawn uniformly randomly from $\{L_i, U_i\}$. This is the most straightforward option, analogous to bit-flipping for binary encoding and the random resetting sketched above for integer encodings. It is normally used with a position form mutation probability Eiben & Smith, 2003).

Non-Uniform Mutation with a Fixed Distribution: Perhaps the most common form of non-uniform mutation used with floating-point representations takes a form analogous to creep mutation for integers. It is designed so that usually but not always the amount of change introduced is small. This is achieved by adding to the current gene value an amount drawn randomly from a Gaussian distribution with mean zero and user specified standard deviation and then curtailing the resulting value to the range $\{L_i, U_i\}$ if necessary. The Gaussian distribution has the property that approximately two thirds of the samples drawn lie within one standard deviation. This means that most of the changes made will be small but there is nonzero probability of generating very large changes since tail of the distribution never reaches zero. It is normal practice to apply this operator with probability one per gene and instead the mutation parameter is used to control the standard deviation of the Gaussian and hence the probability distribution of the step sizes taken (Eiben & Smith, 2003).

Mutation Operators for Permutation Representations: For permutation representations it is no longer possible to consider each gene independently rather finding legal mutations is a matter of moving alleles around in the genome. This has the immediate consequence that the mutation parameter is interpreted as the probability that the string undergoes mutation rather than that a single gene in the string is altered. The three most common forms of mutation used for order-based problems. Whereas the first three operators below work by making small changes to the order in which allele values occur for neighborhood problems these can huge numbers of links to be broken and so inversion is more commonly used (Eiben & Smith, 2003).

• **Swap Mutation:** This operator works by randomly picking two positions (genes) in the string and swapping their allele values. This is illustrated in below, where the values in positions two and five have been swapped.

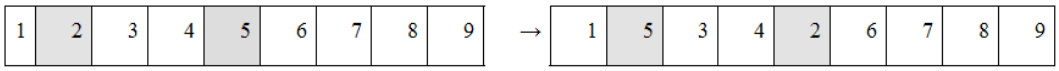

Figure 2.15 Swap mutation (Eiben & Smith, 2003)

• **Insert Mutation:** This operator works by picking two alleles at random and moving one so that it is next to other, shuffling along the others to make room. This is illustrated below, where the values two and five have been chosen.

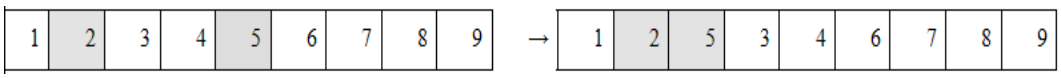

Figure 2.16 Insert mutation (Eiben & Smith, 2003)

• **Inversion Mutation:** Inversion mutation works by randomly selecting two positions in the string and reversing the order in which the values appear between those positions. It effectively breaks the string into three parts with all links inside a part being preserved and only two links between the parts broken.

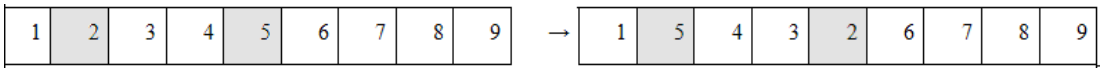

Figure 2.17 Inversion mutation (Eiben & Smith, 2003)

2.2.9 Termination Criteria

Termination is the criterion by which the genetic algorithm decides whether to continue searching or stop the search. Each of the enabled termination criterion is checked after each generation to see if it is time to stop. Some types of termination criteria as follows (Mitchell, 1996; Whitley, 1994):

- **Generation Number**: A termination method that stops the evolution when the user-specified max numbers of evolutions have been run. This termination method is always active.

- **Evolution Time**: Termination method that stops the evolution when the elapsed evolution time exceeds the user-specified max evolution time. By default, the evolution is not stopped until the evolution of the current generation has completed, but this behavior can be changed so that the evolution can be stopped within a generation.

- **Fitness Threshold**: A termination method that stops the evolution when the best fitness in the current population becomes less than the user-specified fitness threshold and the objective is set to minimize the fitness. This termination method also stops the evolution when the best fitness in the current population becomes greater than the user-specified fitness threshold when the objective is to maximize the fitness.

- **Fitness Convergence**: A termination method that stops the evolution when the fitness is deemed as converged. Two filters of different lengths are used to smooth the best fitness across the generations. When the smoothed best fitness from the long filter is less than a user-specified percentage away from the smoothed best fitness

from the short filter, the fitness is deemed as converged and the evolution terminates.

• **Population Convergence**: A termination method that stops the evolution when the population is deemed as converged. The population is deemed as converged when the average fitness across the current population is less than a user-specified percentage away from the best fitness of the current population.

• **Gene Convergence**: A termination method that stops the evolution when a user-specified percentage of the genes that make up a chromosome are deemed as converged. A gene is deemed as converged when the average value of that gene across all of the chromosomes in the current population is less than a user-specified percentage away from the maximum gene value across the chromosomes.

CHAPTER THREE
OUTLIERS AND OUTLIER DETECTION METHODS

In this chapter, it is discussed outliers and outlier detection methods. A variety of outlier detection techniques in databases will be reviewed. In particular, outlier management will be summarized in various aspects including their detection methods, causes and treatments.

3.1 Database Systems

Knowledge of database technology increases in importance day to day. It is a key components of e-commerce and other web-based applications and they lay at the heart of organization-wide operational and decision support applications. Databases also are used by thousands of workgroups and millions of individuals. In fact, estimates of the number of active databases in the world today exceed millions (Kroenke, 2003).

A database management system (DBMS), is software designed to assist in maintaining and utilizing large collections of data, and the need for such systems, as well as their use, is growing rapidly. The alternative to using a DBMS is to use ad hoc approaches that do not carry over from one application to another; for example to store the data and write application-specific code to manage it (Ramakrishnan & Gehrke, 2000).

Several vendors (e.g., IBM, DB, Oracle) have extended their systems with ability to store new data types such as images and text, and with the ability to ask more complex queries. Specialized systems have been developed by numerous vendors for creating data warehouses, consolidating data from several databases, and for carrying out specialized analysis. Database management continues to gain importance as more and more data is brought on-line, and made ever more accessible through computer networking. Today the field is being driven by exciting visions such as multivariate databases and digital libraries (Ramakrishnan & Gehrke, 2000).

Although the most common use of database is arguably to discover relationships or patterns in data with minimal human intervention, an often overlooked but important task is the ability to detect outliers or exceptions in data for the best quality of data. Indeed, for some applications e.g., credit card fraud or telephone calling card fraud, the patterns may be well-established, but it is often the exceptions to those patterns that merit special attention.

3.2 The Quality of Data in Databases

The quality of data sets depend on a number of issues, but the source of the data is crucial factor. Data entry is inherently prone to errors, both simple and complex. Much effort can be allocated to this front-end process with respect to reduction in entry error but the fact often remains that errors in large data set are common. While one can establish a process to obtain high quality data sets, this does not little to address the problem of existing data. The field errors rates in the data acquisition phase are typically around 5% or more (Iglewicz & Hoaglin, 1993).

Caring about data quality is a key to safeguarding and improving it. Discovering whether data are of acceptable quality is a measurement task, and not a very easy one. This observation becomes all the more important in this information age, when explicit and meticulous attention to data is of growing importance if information is not to become misinformation. If data are fit for use in their intended operational, decision making and other roles, data are of high quality (Maronna et. al., 2006).

In many settings, especially for intermediate products, it is also convenient to define quality as Conformance to Standards that have been set, so that fitness for use is achieved. These two criteria link the role of the employee doing work (conformance to standards) to the client receiving the product (fitness for use). When used together, these two can yield efficient systems that achieve the desired accuracy level or other specified quality attributes. Unfortunately, the data of many organizations do not meet either of these criteria (Maronna et. al., 2006).

For existing data sets the logical solution is to attempt to cleanse the data in some way. That is, explore the data set for possible problems and endeavor to correct the errors. A manual process of data cleansing is also laborious, time consuming, and itself prone to errors. Useful and powerful tool that automate or greatly assist in the data cleansing process are necessary and may be the only practical and cost effective way to achieve a reasonable quality level in existing data (Iglewicz & Hoaglin, 1993).

The serious need to store, analyze and investigate such very large data sets has given rise to fields of databases and data mining. Without clean and correct data the usefulness of databases and data mining are mitigated. Thus, data cleansing is a necessary precondition for successful knowledge discovery in databases (Iglewicz & Hoaglin, 1993).

Several areas in computer and other science have in the past treated related and overlapping problems; at the same time, and statistical models and methodologies that have proved to be of major importance in grounding the data quality research area. It is showed that in Figure 3.1 research areas related to data quality (Batini & Scannapieca, 2006).

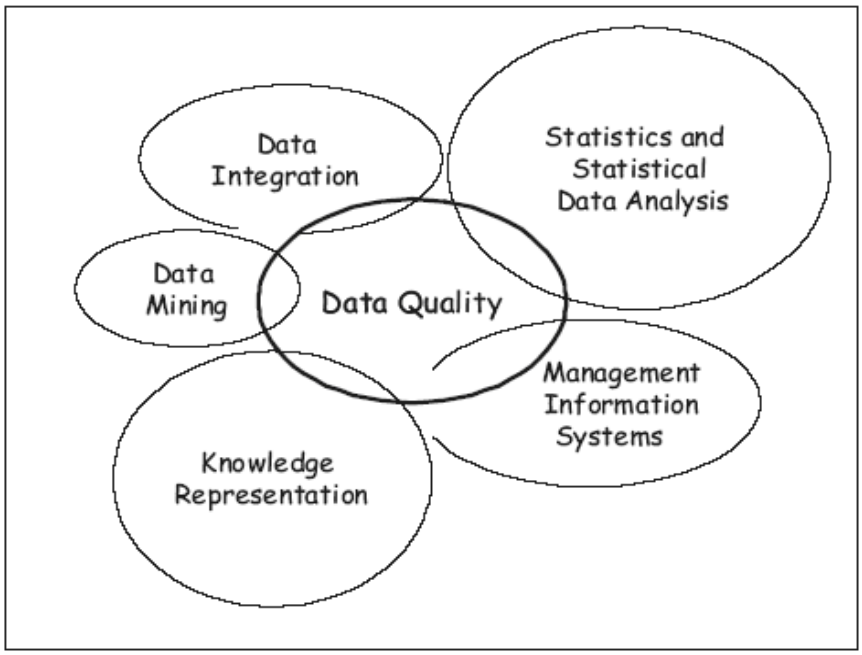

Figure 3.1 Research areas related to data quality (Batini & Scannapieca, 2006)

As seen in Figure 3.1 data quality is an important issue for many areas and data cleansing processes are tied directly to data acquisition and to improve data quality in an existing system. The following three phases define a data cleansing process (Iglewicz & Hoaglin, 1993):

- Define and determine error types,
- Search and identify error instances,
- Correct the uncovered errors.

Each of these phases constitutes a complex problem in itself, and a wide variety of specialized methods and technologies can be applied to each. While data integrity analysis can uncover a number of possible errors in a data set, it does not address more complex error. Errors involving relationships between one or more fields are often very difficult to uncover. These types of errors require deeper inspection and analysis. One can view this as a problem in outlier detection. Simply put: if a large percentage (e.g., 99.9%) of the data elements conform to general form, then the remaining (e.g.,0.1%) data elements are likely error candidates. These data elements are considered outliers. Two things are done here; identifying outliers or strange variations in a data set and identifying trends or normality in data (Iglewicz & Hoaglin, 1993).

Knowing what data is supposed to look like allows errors to be uncovered. However, the fact of the matter is that real world data is often very diverse and rarely conforms to any standard statistical distribution. This problem is especially acute when viewing the data in several dimensions. Therefore, more than one method for outlier detection is often necessary to capture most of outliers. Some of the general methods that can be utilized for error detection are; statistical outlier detection methods for example Chebyshev's theorem, clustering methods, pattern-based methods and association rules. In the next subsection gives information about these and other methods.

3.3 Outliers in Databases

Outliers exist extensively in real world, any they are generated from different sources: a heavily tailed distribution or errors in inputting the data. Mostly, they are “so different from other observations as to arouse suspicions that it was generated by a different mechanism” (Hawking, 1980). According to Barnett and Lewis (1994), an outlier is one that appears to deviate markedly from other members of the sample in which it occurs. Similar definition is pointed out by Beckman and Cook (1983): Observations that stand apart from the bulk of the data are termed as outliers, discordant observations, contaminants, surprising values, or dirty data.

The statistical definition of an outlier depends on the underlying distribution of the variable in question. Thus, Mendenhall et. al., (1993) apply the term outliers to values that lie very far from the middle of the distribution in either direction. This intuitive definition is certainly limited to continuously valued variables having a smooth function of probability density. However, the numeric distance is not the only consideration in detecting continuous outliers. The importance of outlier frequency is emphasized in a slightly different definition, provided by Pyle (1999): “An outlier is a single, or very low frequency, occurrence of the value of a variable that is far away from the bulk of the values of the variable”. The frequency of occurrence should be an important criterion for detecting outliers in categorical data, which is quite common in real-world databases. A more general definition of an outlier is: an observation which appears to be inconsistent with remainder of that set of data.

Outliers may be the result of recording errors or data entry errors, but these may also be legitimate data. In fact, outliers may be point out surprising, suspicious, and fraudulent activities (Knorr, 2002; Pyle, 1999).

The isolation of outliers is important both for improving the quality of original data and for reducing the impact of outlying values in the process in databases. Most existing method of outlier detection is based on manual inspection of graphically represented data (Last & Kandel, 2001). The real cause of outlier occurrence is usually unknown to

data users or analysis. Sometimes, this is a flawed value, resulting from the poor quality of data set i.e., a data entry or a data conversion error. Physical measurements, especially when performed with malfunctioning equipment, may produce a certain amount of distorted values. In these cases, no useful information is conveyed by the outlier value. However, it is possible that an outlier represent correct, though exceptional information. For example, if clusters of outliers result from fluctuations in behavior of a controlled process, their values are important for process monitoring (Last & Kandel, 2001).

Outlier detection purpose of to find the small portion of data which are deviating from common patterns in the database. Studying the extraordinary behavior of outliers helps uncovering the valuable knowledge hidden behind them. The hidden knowledge obtained can be useful in the detection of several areas. Although outliers are commonly measurement or recording errors, some of them can represent event of interest, something significant from the viewpoint of the application domain. Consequently, simply rejecting all outliers may lose useful information, and lead to inaccurate or incorrect results in data analysis tasks. For example, in fraud detection, suspicious credit card transactions may indeed be fraudulent, but, could also be those looking-suspicious, but legitimate ones. In hand-written character recognition, a good outlier might be an atypical but legitimate pattern, while a bad outlier might be a garbage pattern. Isolating outliers may also have a positive impact on the result of data analysis in databases. Simple statistical estimates, like sample mean and standard deviation can be significantly biased by individual outliers that are far away from the middle of the distribution.

In order to judge whether an outlier is informative or useful in practical context, other information is often needed, such as relevant domain or common-sense knowledge, or the experience of data analysts in relation to judging outlying data points, etc. To date, the progress in the explicit management of outliers has been largely restricted to the automated detection but manual analysis of outliers, as in the investigation of credit card fraud and inside dealing at stock markets, in hand-written character recognition, or in the study of customer behavior (Goonatilake et. al.,1995).

Only after the knowledge becomes available and represented in a computable format, is it possible to developed automated methods to prevent the useful outliers from being precluded by statistical evolutionary methods. For example, genetic algorithm will be used for outlier detection.

3.4 Causes of Outliers in Databases

Outliers can have harmful effects on statistical analyses in databases. First, they generally serve to increase error variance and reduce the power of statistical tests. Second, if non-randomly distributed they can decrease normality in multivariate analyses, violate assumptions of sphericity and multivariate normality, altering the odds of making both Type I and Type II errors. Third, they can seriously bias or influence estimates that may be of substantive interest.

Outliers can arise from several different mechanisms or causes. Anscombe (1960) sorts outliers into two major categories: those arising from errors in the data, and those arising from the inherent variability of the data. Not all outliers are illegitimate contaminants, and not all illegitimate scores show up as outliers (Barnett & Lewis, 1994). It is therefore important to consider the range of causes that may be responsible for outliers in a given data set. Reasons of outliers are classified as the follows in (Hair et. al., 1998; Osborne & Overbay, 2004).

- Outliers from data errors: Outliers are often caused by human error, such as errors in data collection, recording, or entry. Data from an interview can be recorded incorrectly upon data entry. Thus, if sufficient information is available, recalculation is a method of saving important data and eliminating an obvious outlier. If outliers of this nature cannot be corrected they should be eliminated as they do not represent valid population data points.

- Outliers from intentional or motivated mis-reporting: There are times when participants purposefully report incorrect data to experimenters or surveyors. Depending on the details of the research, one of two things can happen: inflation of

all estimates, or production of outliers. If all subjects respond the same way, the distribution will shift upward, not generally causing outliers. However, if only a small subsample of the group responds this way to the experimenter, or if multiple researchers conduct interviews, then outliers can be created.

• Outliers from sampling error: Another cause of outliers is sampling. It is possible that a few members of a sample were inadvertently drawn from a different population than the rest of the sample.

• Outliers from standardization failure: Outliers can be caused by research methodology, particularly if something anomalous happened during a particular subject's experience.

• Outliers as legitimate cases sampled from the correct population: Finally, it is possible that an outlier can come from the population being sampled legitimately through random chance. It is important to note that sample size plays a role in the probability of outlying values. Within a normally distributed population, it is more probable that a given data point will be drawn from the most densely concentrated area of the distribution, rather than one of the tails. As a researcher casts a wider network and the data set becomes larger, the more the sample resembles the population from which it was drawn, and thus the likelihood of outlying values becomes greater. In other words, there is only about a 1% chance you will get an outlying data point from a normally-distributed population; this means that, on average, about 1% of your subjects should be 3 standard deviations from the mean.

In the case that outliers occur as a function of the inherent variability of the data, opinions differ widely on what to do. Due to the deleterious effects on power, accuracy, and error rates that outliers can have, it might be desirable to use a transformation or recoding/truncation strategy to both keep the individual in the data set and at the same time minimize the harm to statistical inference.

- Outliers as potential focus of inquiry: Outliers can represent a nuisance, error, or legitimate data. Before discarding outliers, researchers need to consider whether those data contain valuable information that may not necessarily relate to the intended study, but has importance in a more global sense.

In the following sections examined that approaches to detecting outliers, both inside and outside the statistics community. In particular, it is given that graphical methods, distribution- based methods, dept-based methods, basic robust methods, and data clustering algorithms, machine learning, and sequential exceptions.

3.5 Literature Review for Handling Outliers

Outlier detection methods can be divided between univariate and multivariate methods that usually form most of the current body of research. Another fundamental taxonomy of outlier detection methods is between parametric methods and nonparametric methods that are model-free (Williams et al., 2002). Statistical parametric methods either assume a known underlying distribution of the observations (Barnett & Lewis, 1994; Hawkins, 1980; Rousseeuw & Leory, 1987) or, at least, they are based on statistical estimates of unknown distribution parameters (Caussinus & Roiz, 1990; Hadi, 1992). These methods flag as outliers those observations that deviate from the model assumptions. They are often unsuitable for high-dimensional data sets and for arbitrary data sets without prior knowledge of the underlying data distribution (Papadimitriou et al., 2002). Within the class of non-parametric outlier detection methods one can set apart the data mining methods, also called distance-based methods. These methods are usually based on local distance measures and are capable of handling large databases (Bay & Schwabacher, 2003; DuMouchel & Schonlau, 1998; Fawcett & Provost, 1997; Jin et al., 2001; Hawkins et al., 2002; Knorr & Ng, 1997; Williams & Huang ,1997). Another class of outlier detection methods is founded on clustering techniques, where a cluster of small sizes can be considered as clustered outliers (Acuna & Rodriguez, 2004; Hu & Sung 2003), whom proposed a method to identify both high and low density pattern clustering, further partition this class to hard classifiers and soft classifiers.

Traditional studies on detecting outliers lie in the field of statistics, and a number of statistical tests, called discordancy tests are developed (Barnett & Lewis, 1994; Hawkins, 1980). In some practices like monitoring a manufacturing process, a 3σ rule is generally adopted. All these methods are developed to detect a single outlier, and they may fail when multiple outlier exist. Some researches proposed different methods for detecting outliers in multivariate data without the a-priori assumption of the distribution. Knorr and Ng (1998) gave their definition of distance-based (DB) outliers. Ramaswamy et al., (2000) argued that the DB(p,D) outliers are too sensitive to the parameter p and D. They defined a k-nearest neighbor outlier. They calculate the kth nearest distances for all data points and rank the points according to these distances, and then pick the top n as outliers. Breunig et al., (2000) proposed another notion of local outliers. They think that a data point is an outlier which a local neighborhood of the points. They assign each object with an outlier degree, which they call local outlier factor. Thus, they use a continuous score to measure the outlier instead of give the binary result yes or no. Aggarwal et al., (2001) claim about the distance-based and local outliers do not work well for high dimensional dataset since the data are sparse, and outliers should be defined in sub-space projections. They proposed an evolutionary algorithm to find the outliers.

Mentioned these methods are developed to detect individual outliers, and the association of outliers has been studied by Song and Donald (2002). They present an outlier-based data association method. Instead of defining outlier for individual record, they considered to build the outlier measure for a group of data points. These data points are similar on some attributes and are different on other attributes. If these common characteristics are quite unusual, or in other words, they are outliers, these data points are well separated from other points.

Ahmet Kaya (2004) used in his article “Outlier effects in databases”, outlier detection algorithm based model to simulate finding outliers in databases. He was concerned with outliers in time series which have two special cases, innovational outlier (IO) and additive outlier (AO). The occurrence of AO indicates that action is required, possibly to adjust the measuring instrument or at least to print an error message on the database.

Sanjoy Kumar Sinha (1997) suggested a robust sequential procedure for the identification of multiple outliers in multivariate normal data in his master thesis. This thesis deals with the problem of identifying and testing a set of a number k of extreme sample points as significant outliers in a sample of size n drawn from a p-dimensional normal distribution with unknown parameters.

Mark Last and Abraham Kandel presented (2001) automating the process of detecting and isolating outliers. This process was based on modeling the human perception of exceptional values by using the fuzzy set theory.

Another related class of methods consists of detection techniques for spatial outliers. These methods search for extreme observations or local instabilities with respect to neighboring values, although these observations may not be significantly different from the entire population (Lu et al., 2003; Shekhar & Chawla, 2002). A broad review of outlier detection techniques for numeric as well as symbolic data is presented by Agyemang et al., (2006). An extensive review of novelty detection techniques using neural networks and statistical approaches has been presented in Markou and Singh (2003), Patcha and Park (2007) and Snyder (2001) present a survey of outlier detection techniques used specifically for cyber-intrusion detection.

3.6 Classification of Outlier Detection Methods

Outlier detection has been used for centuries to detect and, where appropriate, remove anomalous observations from data. Their detection can identify system faults and fraud before they escalate with potentially catastrophic consequences. It can identify errors and remove their contaminating effect on the data set and as such to purify the data for processing. The original outlier detection methods were arbitrary but now, principled and systematic techniques are used, drawn from the full gamut of computer science and statistics. There are three fundamental approaches to the problem of outlier detection (Victoria & Austin, 2004):

Type 1: Determine the outliers with no prior knowledge of the data. This is essentially a learning approach analogous to unsupervised clustering. The approach processes the data as a static distribution, pinpoints the most remote points, and flags them as potential outliers. Type 1 assumes that errors or faults are separated from the normal data and will thus appear as outliers. There are two sub-techniques commonly employed, diagnosis and accommodation (Rousseeuw & Leroy, 1996). An outlier diagnostic approach highlights the potential outlying points. An alternative methodology is accommodation that incorporates the outliers into the distribution model generated and employs a robust classification method. These robust approaches can withstand outliers in the data and generally induce a boundary of normality around the majority of the data which thus represents normal behavior. In contrast, non-robust classifier methods produce representations which are skewed when outliers are left in.

Type 2: Model both normality and abnormality. This approach is analogous to supervised classification and requires pre-labeled data, tagged as normal or abnormal. A type 2 approach can be used for on-line classification, where the classifier learns the classification model and then classifies new exemplars as and when required against the learned model. If the new exemplar lies in a region of normality it is classified as normal, otherwise it is flagged as an outlier. Classification algorithms require a good spread of both normal and abnormal data, i.e., the data should cover the entire distribution to allow generalization by the classifier.

Type 3: Model only normality or in a very few cases model abnormality (Fawcett & Provost 1999; Japkowicz et al., 1995). Authors generally name this technique novelty detection or novelty recognition. It is analogous to a semi-supervised recognition or detection task and can be considered semi-supervised as the normal class is taught but the algorithm learns to recognize abnormality. The approach needs pre-classified data but only learns data marked normal. It is suitable for static or dynamic data as it only learns one class which provides the model of normality. It can learn the model incrementally as new data arrives, tuning the model to improve the fit as each new exemplar becomes available. It aims to define a boundary of normality.

Most of the existing outlier detection techniques solve a specific formulation of the problem. The formulation is induced by various factors such as nature of the data, availability of labeled data, type of anomalies to be detected, and etc. Often, these factors are determined by the application domain in which the anomalies need to be detected. Researchers have adopted concepts from diverse disciplines such as statistics, machine learning, data mining, information theory, spectral theory, and have applied them to specific problem formulations. Figure 3.2 shows the above mentioned key components associated with any outlier detection technique (Chandola et. al., 2007).

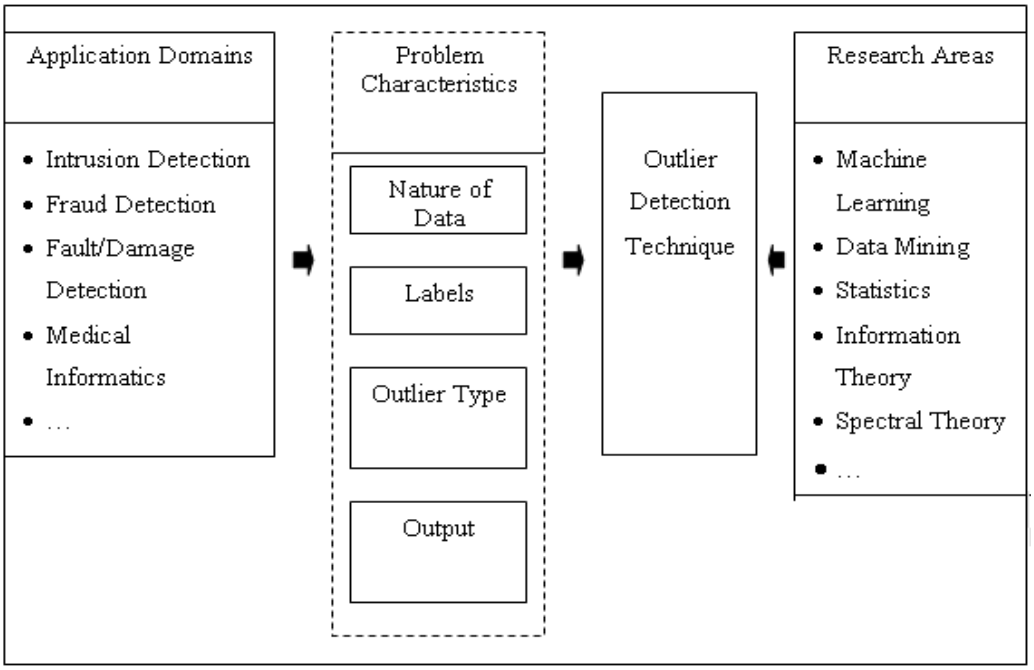

Figure 3.2 Components associated with an outlier detection technique (Chandola et. al., 2007)

Outlier detection has been the topic of a number of surveys and review articles, as well as books. In the literature outlier detection methods can be divided between univariate methods and multivariate methods. One of the above-mentioned classes is seen in Figure 3.3.

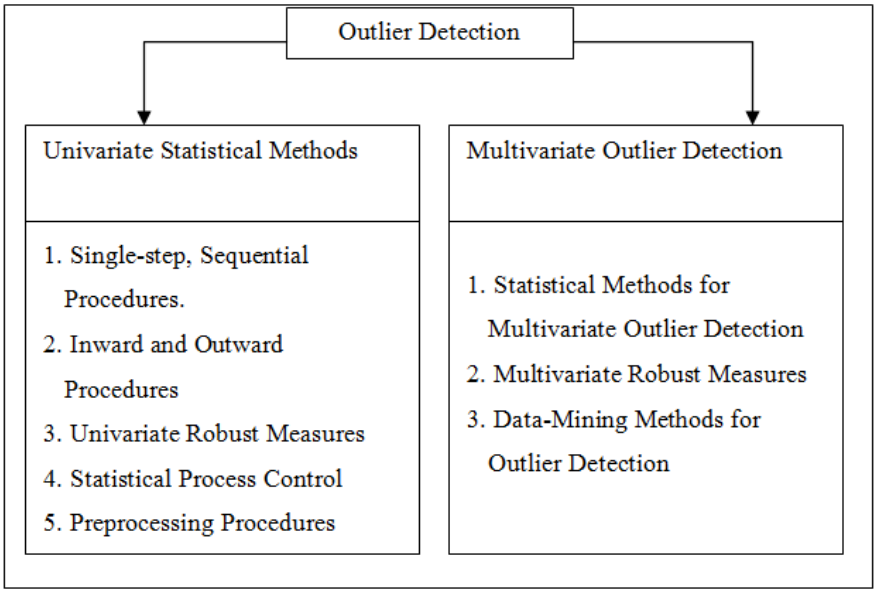

Figure 3.3 Classification for outlier detection

Another fundamental taxonomy of outlier detection methods is between parametric and non-parametric methods. Statistical parametric methods either assume a known underlying distribution of the observations or, at least, they are based on statistical estimates of unknown distribution parameters. These methods flag as outliers those observations that deviate from the model assumptions. They are often unsuitable for high dimensional data sets and for arbitrary data sets without prior knowledge of the underlying data distribution. Within the class of non-parametric outlier detection methods one can set apart the data mining techniques, also called distance based methods. These methods are usually based on local distance measures and capable of handling large databases. A class of outlier detection method is founded on clustering techniques, where a cluster of small sizes can be considered as clustered outliers.

Barnett & Lewis (1994) and Rousseeuw & Leroy (1996) describe and analyze a broad range of statistical outlier techniques and Marsland (2001) analyses a wide range of neural methods. Victoria and Austin (2004) provided an extensive survey of outlier detection techniques developed in machine learning and statistical domains. Their study is categorized and analyzed broad range of outlier detection methodologies. They pinpoint how each handles outliers and make recommendations for when each methodology is appropriate for clustering, classification and recognition. They have

observed that outlier detection methods are derived from four fields of computing: statistics (proximity-based, parametric, non-parametric and semi-parametric), neural networks (supervised and unsupervised) and machine learning. Victoria and Austin (2004) classes is seen in Figure 3.4.

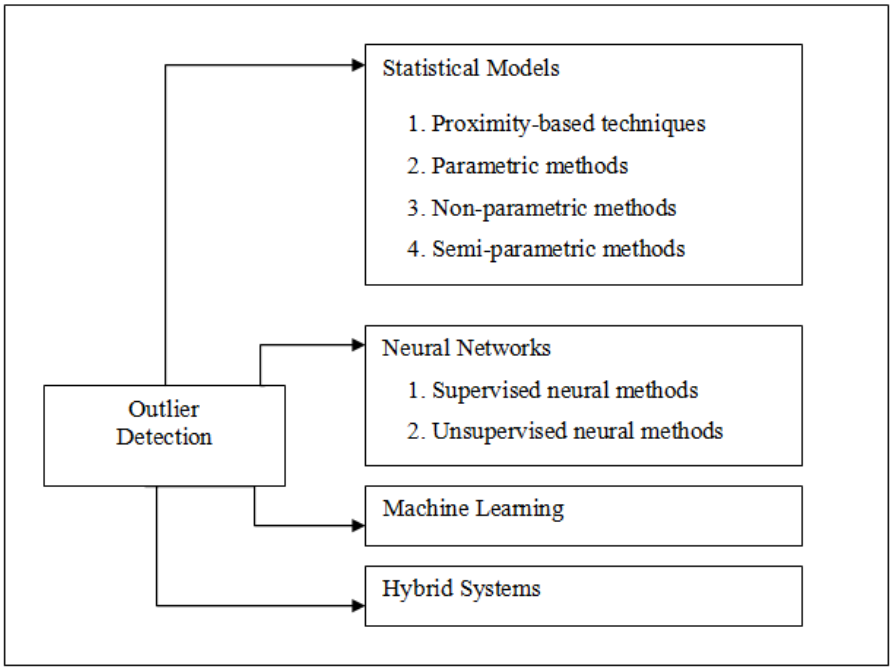

Figure 3.4 Classification for outlier detection (Victoria & Austin, 2004)

Chandola et. al., (2007) attempt to provide a structured and a broad overview of extensive research on outlier detection techniques spanning multiple research areas and application domains. Most of the existing surveys on outlier detection either focus on a particular application domain or on a single research area. Agyemang et al., (2006) and Victoria & Austin (2004) are two related works that group outlier detection into multiple categories and discuss techniques under each category. Chandola et. al., (2007) add two more categories of outlier detection techniques, information theoretic and spectral techniques, to the four categories discussed in Agyemang et al., (2006) and Victoria & Austin (2004). For each of the six categories, Chandola et. al., (2007) not only discusses the techniques, but also identify unique assumptions regarding the nature of outlier made by the techniques in that category. These assumptions are critical for determining when the techniques in that category would be able to detect outliers, and when they would fail. For each category, Chandola et. al., (2007) provide a basic outlier detection

technique, and then show how the different existing techniques in that category are variants of the basic technique. This template provides an easier understanding of the techniques belonging to each category, and also, for each category are identified the advantages and disadvantages of the techniques. They provide a discussion on the computational complexity of the techniques since it is an important issue in real application domains. Chandola et. al., (2007) classes for outlier detection is seen in Figure 3.5.

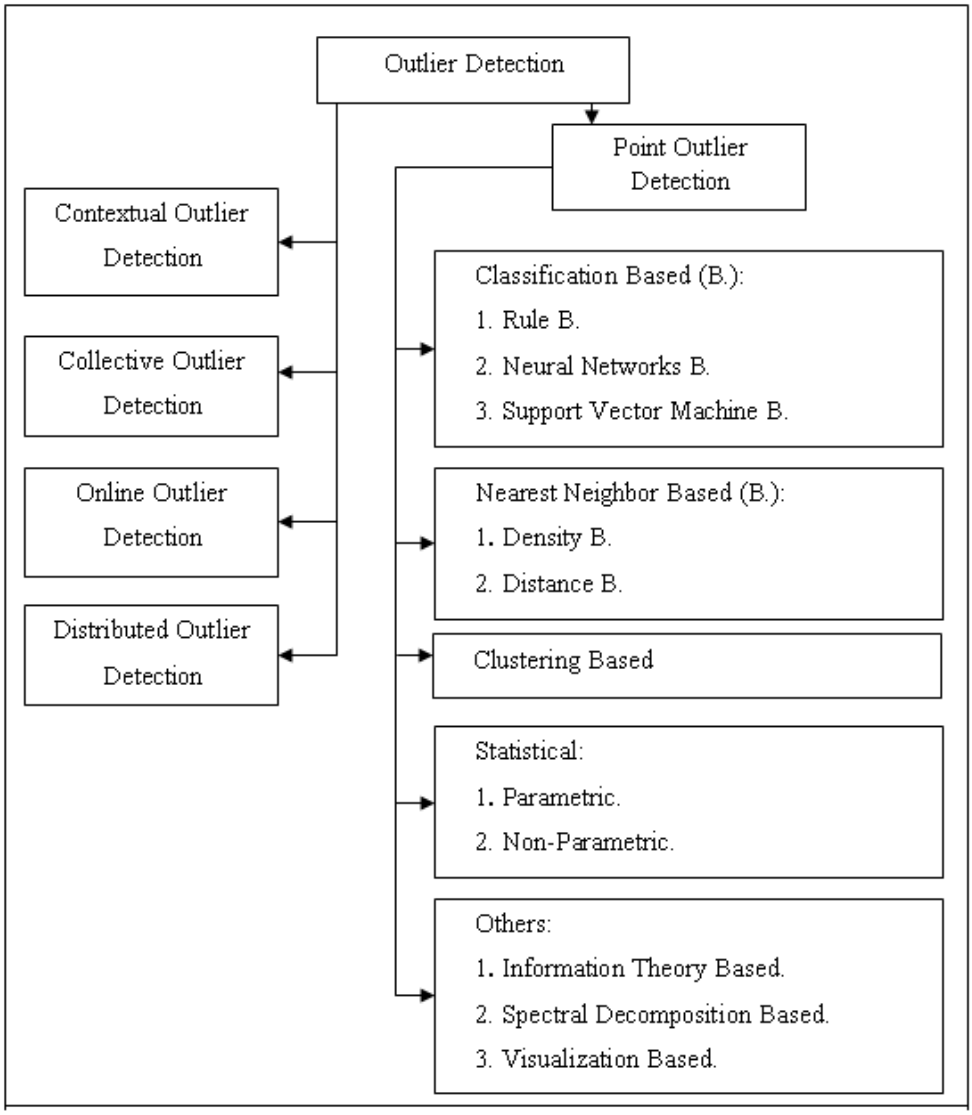

Figure 3.5 Classification for outlier detection (Chandola et. al., 2007)

An important aspect of an outlier detection technique is the nature of the desired outlier. According to Chandola et. al., (2007) outliers can be classified into following three categories:

I. Point Outlier: If an individual data instance can be considered as outlier with respect to the rest of data, then the instance is termed as a point anomaly. This is the simplest type of outlier and is the focus of majority of research on anomaly detection.

For example, in Figure 3.6 points o1 and o2 as well as points in region O3 lie outside the boundary of the normal regions, and hence these are point anomalies since they are different from normal data points.

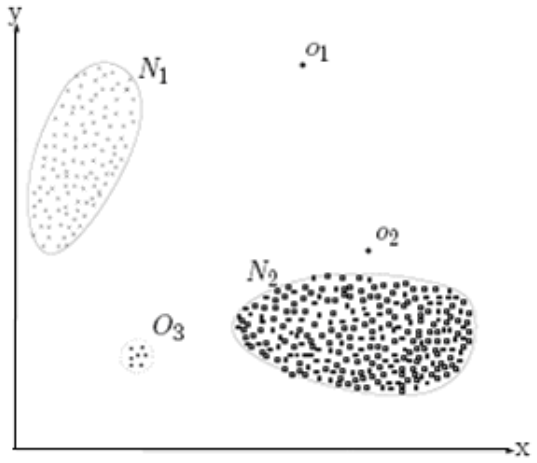

Figure 3.6 An Example of outliers in a 2 dimensional data set (Chandola et. al., 2007)

As a real life example, consider credit card fraud detection. Let the data set correspond to an individual's credit card transactions. For the sake of simplicity, let us assume that the data is defined using only one feature: amount spent. A transaction for which the amount spent is very high compared to the normal range of expenditure for that person will be a point outlier.

II. Contextual Outlier: If a data instance is anomalous in a specific context, then it is termed as a contextual anomaly also referred to as conditional anomaly (Song et al., 2007). The notion of a context is induced by the structure in the data set and has to be specified as a part of the problem formulation. Each data instance is defined using

contextual and behavioral attributes: The contextual attributes are used to determine the context or neighborhood for that instance. For example, in spatial data sets, the longitude and latitude of a location are the contextual attributes. In time series data, time is a contextual attribute which determines the position of an instance on the entire sequence. The behavioral attributes define the non-contextual characteristics of an instance. For example, in a spatial data set describing the average rainfall of the entire world, the amount of rainfall at any location is a behavioral attribute (Chandola et. al., 2007).

Contextual outliers have been most commonly explored in time-series data (Salvador & Chan 2003; Weigend et al., 1995) and spatial data (Kou et al., 2006; Shekhar et al., 2001). Figure 3.7 shows one such example for a temperature time series which shows the monthly temperature of an area over last few years. A temperature of 35F might be normal during the winter (at time t_1) at that place, but the same value during summer (at time t_2) would be an outlier (Chandola et. al., 2007).

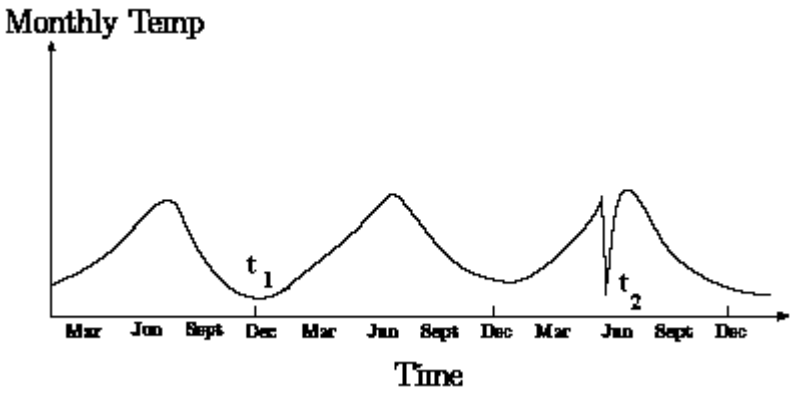

Figure 3.7 Contextual outlier* (Chandola et. al., 2007)

III. Collective Outlier: If a collection of related data instances is anomalous with respect to the entire data set, it is termed as a collective outlier. The individual data instances in a collective anomaly may not be anomalies by themselves, but their occurrence together as a collection is anomalous. Figure 3.8 illustrates an example which shows a human electrocardiogram output (Goldberger et al., 2000). The

* t_2 in a temperature time series. Note that the temperature at time t_1 is same at that at time t_2 but occurs in a different context and hence is not considered as an outlier.

highlighted region denotes an anomaly because the same low value exists for an abnormally long time corresponding to an Atrial Premature Contraction. Note that, that low value by itself is not an outlier (Chandola et. al., 2007).

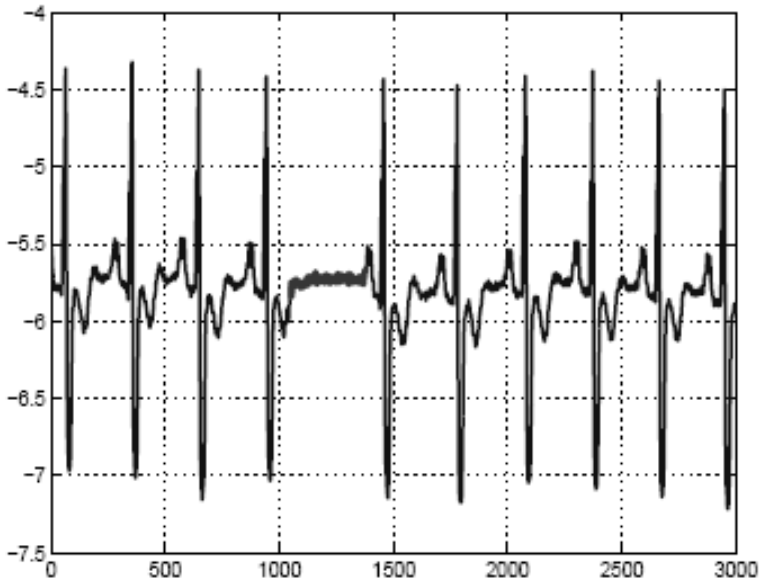

Figure 3.8 Collective anomalies[†] (Chandola et. al., 2007)

It should be noted that while point outliers can occur in any data set, collective outliers can occur only in data sets in which data instances are related. In contrast, occurrence of contextual outliers depends on the availability of context attributes in the data. A point anomaly or a collective anomaly can also be a contextual anomaly if analyzed with respect to a context. Thus a point outlier detection problem or collective outlier detection problem can be transformed to a contextual outlier detection problem by incorporating the context information (Chandola et. al., 2007).

The existing surveys mention the different applications of outlier detection and it is provided a detailed discussion of the application domains which are mentioned. In the next subsections, it is described some of the above mentioned techniques are explained below.

[†] Corresponding to an atrial premature contraction in an human electrocardiogram output.

3.6.1 Statistical Methods for Outlier Detection

Statistical approaches are the earliest methods for outlier detection. In fact, many of the techniques were described in both Barnett & Lewis (1994) and Rousseeuw & Leroy (1996).

Statistical methods can be used to summarize or describe a collection of data. These techniques fit a statistical model to the given data using parametric or non-parametric methods and then apply a statistical inference test to determine if an unseen instance belongs to this model or not. In addition, patterns in the data may be modeled in a way that accounts for randomness and uncertainty in the observations, and are then used to draw inferences about the process.

Parametric and non-parametric techniques have been applied to fit a statistical model. While parametric techniques assume the knowledge of underlying distribution and estimate the parameters from the given data (Eskin, 2000), non-parametric techniques do not generally assume knowledge of underlying distribution (Desforges et. al., 1998). In the next two subsections it is discussed parametric and non-parametric outlier detection techniques.

3.6.1.1 Parametric Methods for Outlier Detection

Parametric methods allow the model to be evaluated very rapidly for new instances and are suitable for large data sets; the model grows only with model complexity not data size. However, they limit their applicability by enforcing a pre-selected distribution model to fit the data. If the user knows their data fits such a distribution model then these approaches are highly accurate but many data sets do not fit one particular model (Victoria & Austin, 2004).

I. Distribution Based Methods: If a model cannot provide an adequate fit or a statistical explanation for an outlier that is not to say that a better model will be found. For example, measurement and recording errors are common causes of outliers, and

these outliers cannot necessarily be explained by a better model. Distribution based methods have been developed for different circumstances, depending on:

- the data distribution,
- whether or not the distribution parameters are known,
- the number of expected outliers , and even
- the types of expected outliers.

However, these methods suffer from the following two serious problems. These tests may not be well-suited to large datasets. First, almost of them are univariate (i.e., single attribute). This restriction makes them unsuitable for multidimensional datasets. Second, all of them are distribution-based, one or more parameters of the distributions are unknown. In numerous situations where we do not know whether a particular attribute follows a normal distribution, a gamma distribution, and so on, we have to perform extensive testing to find a distribution that fits the attribute.

One such single dimensional method is Grubbs' method (extreme studentized deviate) (Grubbs, 1969) which calculates a Z value as the difference between the mean value for the attribute and the query value divided by the standard deviation for the attribute where the mean and standard deviation are calculated from all attribute values including the query value.

II. Depth Based Methods: Ruts and Rousseeuw (1996) proposed a depth based method to detect the outliers. The data points are organized in layers in data space according to the value of the point depth. The depth of a point p to a one–dimensional dataset $X = x_1, x_2, ..., x_n$ is the minimum of the number of data points to the left of p and the number of data points to the right of p. The depth of a d-dimensional data point $p = (p_1, p_2, ..., p_d)$ is defined as the smallest depth of p_i in the i-dimensional projection of the dataset, where $1 \le i \le d$. Outliers are expected to be in the layers with smaller depth. Such methods do not have the distribution fitting problem. Peeling and depth contours are two different notions of depth studied. These depth-based methods avoid the previously mentioned problem of distribution fitting, and conceptually allow

multidimensional data objects to be processed. However, in practice, the computation of d-dimensional layers relies on the computation of d-dimensional convex hulls. Because the lower bound complexity of computing a d-dimensional convex hull for N data objects is ($N^{[d/2]}$), depth-based methods are not expected to be practical for more than four dimensions for large datasets. In fact, existing depth-based methods only give acceptable performance for dimension $k \leq 2$.

Rousseeuw and Leroy (1996) approached the minimum volume ellipsoid estimation (MVE) which fits the smallest permissible ellipsoid volume around the majority of the data distribution model (generally covering 50% of the data points). This represents the densely populated normal region shown in Figure 3.6 (with outliers shown) and Figure 3.9 (with outliers removed).

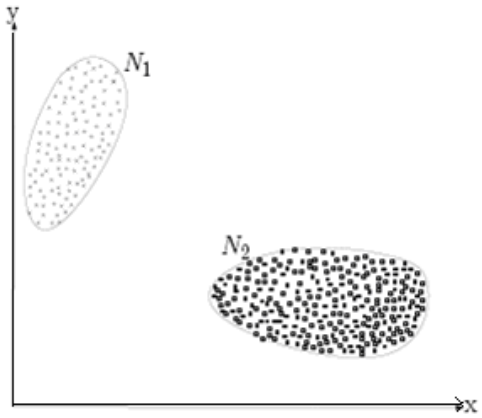

Figure3.9 A data distribution classified by type 3 outlier recognition

III. Regression Model Based: The basic regression model based outlier detection technique consists of two steps. In the first step, a regression model is fitted on the data. In the second step, for each test instance, the residual for the test instance is used to determine the anomaly score. The residual is the part of the instance which is not explained by the regression model. The magnitude of the residual can be used as the anomaly score for the test instance, though statistical tests have been proposed to determine outliers with certain confidence information (Chandola et. al., 2007).

Torr and Murray (1993) proposed peel away outlying points by iteratively pruning and re-fitting. They measure the effect of deleting points on the placement of the least squares standard regression line for a diagnostic outlier detector. The LS line is placed to minimize equation (3.1):

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.1}$$

Torr and Murray's method (1993) repeatedly deletes the single point with maximal influence (the point that causes the greatest deviation in the placement of the regression line) thus allowing the fitted model to stay away from the outliers. They refit the regression to the remaining data until there are no more outliers, i.e., the next point with maximal influence lies below a threshold value. Least squares regression is not robust as the outliers affect the placement of the regression line so it is best suited to outlier diagnostics where the outliers are removed.

Robust regression provides an alternative to least squares regression that works with less restrictive assumptions. Outliers violate the assumption of normally distributed residuals in the least squares regression. Robust regression methods are Least Median of Squares (LMS), Least Trimmed Squares (LTS), Huber M Estimation, MM Estimation, Least Absolute Value Method (LAV) and S Estimation.

3.6.1.2 Non-Parametric Methods for Outlier Detection

The outlier detection techniques in this category use non-parametric statistical models, such that the model structure is not defined a priory, but is instead determined from given data. Such techniques typically make fewer assumptions regarding the data, such as smoothness of density, when compared to parametric techniques.

I. Graphical Methods: Laurikkala et al., (2000) used informal box plots to pinpoint outliers in both univariate and multivariate data sets. This produces a graphical representation (see Figure 3.10 for an example box plot) and allows a human auditor to

visually pinpoint the outlying points. Outliers can be detected graphically in up to 3 dimensions. For the univariate case (1-D), observations can simply be plotted on a line and outliers can be visually detected as those points that lie noticeably apart from the rest of the data.

A box-plot can also be used to display univariate data/outliers. An example is shown in Figure 3.10 and an explanation of its properties is as follows. The box is defined by 3 long horizontal line segments which mark the upper quartile, median and lower quartile; this covers 50% of the data. A vertical line segment extends from the upper quartile to the most extreme observation that is within a distance of (1.5* interquantile range) of the upper quartile (and similarly for the lower quartile). Short horizontal line segments simply denote the ends of these ranges. Finally, any observations lying outside of the range delimited by these short horizontal line segments are marked with a dot: such observations are outliers.

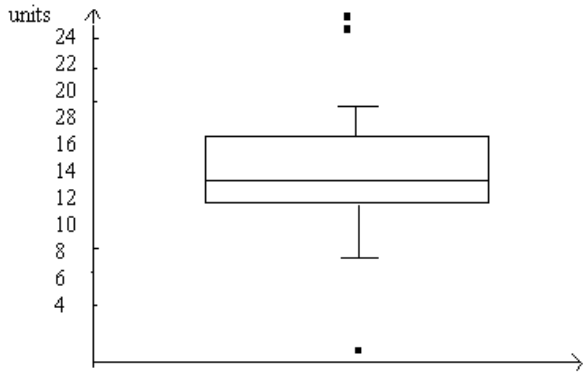

Figure 3.10 Boxplot graph

The box-plot in Figure 3.10 shows that the interquantile is 15-12.5=2.5, the median is 13.5 the rest of the data (except outliers) lies between 8 and 28, the interquantile range skewed slightly to the higher values, and there are 3 outliers.

For visualization in 2-D, a scatterplot can be used and for 3-D a spin plot can be used. Note that, except for boxplots, the detection of outliers using graphical methods can be quite subjective. Disadvantages of graphical methods give limited information about relative strength of an outlier.

II. Histogram Based: The simplest non-parametric statistical technique is to use histograms to maintain a profile of the normal data. Such techniques are also referred to as frequency based or counting based. Histogram based techniques are particularly popular in detection community (Eskin, 2000). These techniques require normal data to build the histograms (Anderson et al., 1994). For multivariate data, a basic technique is to construct attribute wise histograms. The basic histogram based technique for multivariate data has been applied to system call intrusion detection (Endler, 1998), network intrusion detection (Yamanishi et al., 2004).

III. Kernel Function Based. Kernel-based methods estimate the density distribution of the input space and identify outliers as lying in regions of low density. Roberts & Tarassenko (1994) and Bishop (1994) use Gaussian mixture models (GMM) to learn a model of normal data by incrementally learning new exemplars. The GMM is represented by equation (3.2),

$$p(t|x) = \sum_{j=1}^{M} \alpha_j(x)\varphi_j(t|x) \qquad (3.2)$$

where M is the number of kernels $\varphi, \alpha_j(x)$ the mixing coefficients, x the input vector and t the target vector. Roberts and Tarassenko (1994) classify EEG signatures to detect abnormal signals which represent medical conditions such as epilepsy. In both approaches, each mixture represents a kernel whose width is autonomously determined by the spread of the data. In Bishop's approach the number of mixture models is determined using cross-validation. This technique adds new mixture models incrementally. If the mixture that best represents the new exemplar is above a threshold distance, then the algorithm adds a new mixture.

3.6.2 Nearest Neighbor Based Methods for Outlier Detection

These methods are simple to implement and make no prior assumptions about the data distribution model. They are suitable for both type 1 and type 2 outlier detection.

Nearest neighbor based anomaly detection techniques require a distance or similarity measure defined between two data instances. Distance or similarity between two data instances can be computed in different ways. For continuous attributes, Euclidean distance is a popular choice but other measures can be used (Tan et al., 2005). For categorical attributes, simple matching coefficient is often used but more complex distance measures can be used (Boriah et al., 2008; Chandola et al., 2008). For multivariate data instances, distance or similarity is usually computed for each attribute and then combined (Tan et al., 2005).

According to Chandola et. al., (2007) nearest neighbor based anomaly detection techniques can be broadly grouped into two categories:

- Techniques that use the distance of a data instance to its k^{th} nearest neighbor as the anomaly score.
- Techniques that compute the relative density of each data instance to compute its anomaly score.

3.6.2.1 Distance Based Methods for Outlier Detection

A basic nearest neighbor anomaly detection technique is based on the following definition: The anomaly score of a data instance is defined as its distance to its k^{th} nearest neighbor in a given data set. The neighbors are taken from a set of objects for which the correct classification (or, in the case of regression, the value of the property) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. In order to identify neighbors, the objects are represented by position vectors in a multidimensional feature space. It is usual to use the Euclidean distance, though other distance measures, such as the Manhattan distance could in principle be used instead. The k-nearest neighbor algorithm is sensitive to the local structure of the data. Usually Euclidean distance is used as the distance metric; however this will only work with numerical values. In cases such as text classification another metric, such as the overlap metric (or Hamming distance) can be used. The best choice of k depends upon the data; generally, larger values of k reduce the effect of noise on the

classification, but make boundaries between classes less distinct. A good k can be selected by various heuristic techniques, for example, cross-validation.

Distance Based Methods can be explain as: Let N be the number of objects in dataset T, and let F be the underlying distance function that gives the distance between any pair of objects in T. For a object O, the neighborhood of O contains the set of objects $Q \in T$ that are within distance D of O (i.e$\{Q \in T : F(O,Q) \leq D\}$) .The fraction p is the minimum fraction of objects in T that must be outside the D-neighborhood of an outlier. For simplicity of discussion, let M be the maximum allowable number of objects within the D-neighborhood of an outlier (i.e., M=N(1-p)).

From the formulation above it is obvious that given p and D the problem of finding DB(p,D) outliers can be solved by answering a nearest neighbor range query centered at each object O. More specifically based on a standard multidimensional indexing structure, it can execute a range search with radius D for each object O. As soon as we find (M+1) neighbors in the D neighborhood, it is stopped the search and declare O a non-outlier. Otherwise it is reported O as an outlier.

Several variants of the basic technique have been proposed to improve the efficiency. Some techniques prune the search space by either ignoring instances that cannot be anomalous or by focusing on instances that are most likely to be anomalous. Bay and Schwabacher (2003) show that for a sufficiently randomized data, a simple pruning step could result in the average complexity of the nearest neighbor search to be nearly linear. After calculating the nearest neighbors for a data instance, the algorithm sets the anomaly threshold for any data instance to the score of the weakest anomaly found so far. Using this pruning procedure, the technique discards instances that are close, and hence not interesting.

Wu and Jermaine (2006) use sampling to improve the efficiency of the nearest neighbor based technique. The authors compute the nearest neighbor of every instance within a smaller sample from the data set.

Many nearest neighbor search algorithms have been proposed over the years; these generally seek to reduce the number of distance evaluations actually performed. Some optimizations involve partitioning the feature space, and only computing distances within specific nearby volumes. Several different types of nearest neighbor finding algorithms include: k-Most Similar Neighbor (k-MSN), Linear scan, Kd-trees, Balltrees, Metric trees, Locality Sensitive Hashing (LSH), Agglomerative-Nearest-Neighbor, Redundant Bit Vectors (RBV).

Knorr et al., (1998) comment that on two contemporary studies for identifying exceptions in large, multidimensional datasets. The techniques employed are significantly different from those of DB outliers, yet the spirit is similar: the anomalies (outliers) that are identified are intuitively surprising.

Ramaswamy et al., (2000) modified the definition of outlier introduced by Knorr and Ng and consider as outliers the top n points p whose distance to their k^{th} nearest neighbor is greatest. To detect outliers, a partition-based algorithm is presented that, first, partitions the input points using a clustering algorithm and, then, prunes those partitions that can not contain outliers. The experiments, up to 10 dimensions, show that the method scales well with respect to both data size and dimensionality.

Fabrizio Angiulli et al., (2006) introduced the concept of outlier detection solving set S, a subset of D that includes a sufficient number of points from D to allow that to consider only the distances among the pairs in SxD to obtain the top n outliers. They presented an algorithm that computes the solving set and obtained the top n outliers in D and the weight w^* by avoiding to calculate the distance of an object to each other to obtained its k nearest neighbors. Finally, they show that the solving set S, besides containing the top n outliers in D, it allowed to effectively classify each new unseen object as outlier or not by approximating its weight with regard to D with its weight with regard to S.

3.6.2.2 Density Based Methods

Density based anomaly detection techniques estimate the density of the neighborhood of each data instance. An instance that lies in a neighborhood with low density is declared to be anomalous while an instance that lies in a dense neighborhood is declared to be normal. Density based techniques perform poorly if the data has regions of varying densities. For example, consider a two dimensional data set shown in Figure 3.11. Due to the low density of the cluster C1 it is apparent that for every instance q inside the cluster C1, the distance between the instance q and its nearest neighbor is greater than the distance between the instance p_2 and the nearest neighbor from the cluster C2, and the instance p_2 will not be considered as anomaly. Hence, the basic technique will fail to distinguish between p_2 and instances in C1. However, the instance p_1 may be detected Chandola et. al., (2007).

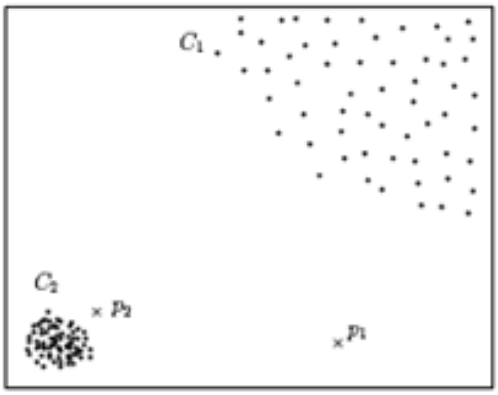

Figure 3.11 Advantage of local density based techniques over global density based techniques (Chandola et. al., 2007)

Breunig et al., 2000 assign an anomaly score to a given data instance, known as Local Outlier Factor (LOF). For any given data instance, the LOF score is equal to ratio of average local density of the k nearest neighbors of the instance and the local density of the data instance itself. To find the local density for a data instance, the authors first find the radius of the smallest hyper-sphere centered at the data instance that contains its k nearest neighbors. The local density is then computed by dividing k by the volume of this hyper-sphere. For a normal instance lying in a dense region, its local density will be similar to that of its neighbors, while for an anomalous instance, its local density will be

lower than that of its nearest neighbors. Hence the anomalous instance will get a higher LOF score. In the example shown in Figure 3.11, LOF will be able to capture both anomalies (p_1 and p_2) due to the fact that it considers the density around the data instances.

3.6.3 Clustering Based Methods

Clustering algorithms assign similar objects in a dataset to the same classes. The algorithms can be divided into two broad categories: partitioning and hierarchical. In both cases, each data object is assigned to at least one cluster. Hierarchical clustering algorithms perform pile and division on the underlying dataset, with disadvantage that the agglomeration and division efforts can not be reversed. Partitioning algorithms, on the other hand, permit points to go back and forth among clusters k, if k is not known.

A few clustering algorithms, such as CLARANS, DBSCAN, and BIRC, are developed with exception-handling capabilities. Outliers are those points that are sufficiently far from a cluster, but they may appear on the border of a cluster. However, their main objective is to find clusters in the dataset. As such, their notions of outliers are defined indirectly through the notion of clusters, and they are developed only to optimize clustering, but not to optimize outlier detection. While it is true that some of the objects which do not form clusters are outliers, it is also true that some of the clusters may contain points that are outliers. Furthermore, one or more relatively small clusters that are separate from the main body of data may be totally comprised of outliers. Outliers actually serve as a bridge between two unrelated clusters inadvertently turning those two clusters into a single cluster.

Several variations of the clustering of techniques have been proposed (Eskin et al., 2002; He et. al. 2003; Pires & Santos-Pereira, 2005; Otey et. al., 2003). The technique proposed by He et al., (2003), called Find Cluster-Based Local Outlier Factor (CBLOF), assigns an anomaly score known as for each data instance. The CBLOF score captures the size of the cluster to which the data instance belongs, as well as the distance of the data instance to its cluster centroid.

3.6.4 Classification Based Methods for Outlier Detection

A classifier that can distinguish between normal and anomalous classes can be learnt in the given feature space. Classification is used to learn a model (classifier) from a set of labeled data instances (training) and then, classify a test instance into one of the classes using the learnt model (testing) (Tan et al., 2005). Classification based anomaly detection techniques operate in a similar two-phase fashion. The training phase learns a classifier using the available labeled training data. The testing phase classifies a test instance as normal or anomalous using the classifier. A variety of outlier detection techniques which are use different classification algorithms to build classifiers as following;

I. Rule Based Technique: The rule-based module may be either a classifier learning classification rules from both normal and abnormal training data or a recognizer trained on normal data only or learning rules to pinpoint changes that identify fraudulent activity. Rule based techniques have been applied in multi-class as well as one-class setting. A basic multi-class rule based technique consists of two steps. First step is to learn rules from the training data using a rule learning algorithm, such as RIPPER, Decision Trees, and etc. Each rule has an associated confidence value which is proportional to ratio between the number of training instances correctly classified by the rule and the total number of training instances covered by the rule. Second step is to find, for each test instance, the rule that best captures the test instance. The inverse of the confidence associated with the best rule is the anomaly score of the test instance. Several minor variants of the basic rule based technique have been proposed (Salvador & Chan, 2003).

II. Neural Network Based Technique: Neural network approaches are generally non-parametric and model based, they generalize well to unseen patterns and are capable of learning complex class boundaries. After training the neural network forms a classifier. However, the entire data set has to be traversed numerous times to allow the network to settle and model the data correctly. They also require both training and testing to fine tune the network and determine threshold settings before they are ready

for the classification of new data. Many neural networks are susceptible to the curse of dimensionality though less so than the statistical techniques. The neural networks attempt to fit a surface over the data and there must be sufficient data density to discern the surface. Most neural networks automatically reduce the input features to focus on the key attributes. But nevertheless, they still benefit from feature selection or lower dimensionality data projections (Victoria & Austin, 2004).

Several variants of the basic neural network technique have been proposed that use different types of neural networks, Multi Layered Perceptron (MLP), Neural Trees, Auto-associative Networks, Adaptive Resonance Theory Based, Radial Basis Function Based, Hopfield Networks, Oscillatory Networks. The MLP interpolate well but perform poorly for extrapolation so cannot classify unseen instances outside the bounds of the training set. It is trained by minimizing the square error between the actual value and the MLP output value given by equation (3.3):

$$\text{Error} = \sum_{j=1}^{m} \int \left[y_j(x;w) - (t_j | x) \right]^2 p(x) d(x) + \sum_{j=1}^{m} \int \left[(t_j^2 | x) - (t_j | x) \right]^2 p(x) d(x) \tag{3.3}$$

from Bishop (1994) where t_j is the target class, y_j is the actual class, $p(x)$ is the unconditional probability density and $y_j(x;w)$ is the function mapping.

Supervised networks require a preclassified data set to permit learning. If this preclassification is unavailable then an unsupervised neural network is desirable. Unsupervised neural networks contain nodes which compete to represent portions of the data set. As with perceptron based neural networks, decision trees or k-means, they require a training data set to allow the network to learn. They autonomously cluster the input vectors through node placement to allow the underlying data distribution to be modeled and the normal/abnormal classes differentiated. They assume that related vectors have common feature values and rely on identifying these features and their values to topologically model the data distribution. Self organizing maps (SOM),

Kohonen, (1997) are competitive, unsupervised neural networks. SOMs perform vector quantization and non-linear mapping to project the data distribution onto a lower dimensional grid network whose topology needs to be pre-specified by the user. Each node in the grid has an associated weight vector analogous to the mean vector representing each cluster in a k-means system. The network learns by iteratively reading each input from the training data set, finding the best matching unit, updating the winner's weight vector to reflect the new match like k-means. However, the SOM also updates the neighboring nodes around the winner unlike k-means.

III. Support Vector Machines (SVM): Support vector machines for type 2 classification which use linear models to implement complex class boundaries. They project the input data onto higher dimensional kernels using a kernel function in an attempt to find a hyper-plane that separates normal and abnormal data. Support vectors functions are positive in the dense regions of the data distribution and negative in the sparsest regions of the distribution where the outliers lie. A support vector function is defined by equation (3.4):

$$SV = \text{sign}\left(\sum_{j=1}^{n} \alpha_j L_j K(x_j, z) + b \right) \tag{3.4}$$

where K is the Kernel function, sign is a function returning +1 if the data is positive and -1 if the data is negative, L_j is the class label, b is the bias, z the test input and x_j the trained input. The data points that define the class boundary of normality are the support vectors. Only this small set of support vectors need be stored often less than 10% of the training set so a large data set can effectively be stored using a small number of exemplars.

A variant of the basic technique (Tax, 2001) finds the smallest hyper-sphere in the kernel space, which contains all training instances, and then determines which side of that hyper-sphere does a test instance lie. If a test instance lies outside the hyper-sphere, it is declared to be anomalous. Song et al., (2002) use Robust Support Vector Machines

(RSVM) which are robust to the presence of anomalies in the training data. RSVM have been applied to system call intrusion detection (Hu et al., 2003).

3.6.5 Other Methods for Outlier Detection

Different paradigms were suggested to improve the efficiency of various data analysis tasks including outlier detection. One possibility is to reduce the size of the data set by assigning the variables to several representing groups. Another option is to eliminate some variables from the analyses by methods of data reduction, such as methods of principal components and factor analysis. Another means to improve the accuracy and the computational tractability of multiple outlier detection methods is the use of biased sampling. Kollios et al., (2003) investigate the use of biased sampling according to the density of the data set to speed up the operation of general data-mining tasks, such as clustering and outlier detection.

Information theoretic techniques analyze the information content of a data set using different information theoretic measures such as Kolmogorov Complexity, entropy, relative entropy, etc. Such techniques are based on the following key assumption: anomalies in data induce irregularities in the information content of the data set. The advantages of information theoretic techniques are as follows Chandola et. al., (2007):

- They can operate in an unsupervised setting.

- They do not make any assumptions about the underlying statistical distribution for the data.

The disadvantages of information theoretic techniques are as follows:

- The performance of such techniques is highly dependent on the choice of the information theoretic measure. Often, such measures can detect the presence of anomalies only when there are significantly large numbers of anomalies present in the data.

- Information theoretic techniques applied to sequences and spatial data sets rely on the size of the substructure, which is often nontrivial to obtain.

- It is difficult to associate an anomaly score with a test instance using an information theoretic technique.

CHAPTER FOUR
INFORMATION CRITERIA

In recent years, the statistical literature has placed increasing emphasis on model-evaluation criteria. The problem is posed as the choice of the best approximating model among a class of competing models by suitable model evaluation criteria given a data set. Model evaluation criteria are figures of merit, or performance measures, for competing models. That model which optimizes the criterion is chosen to be the best (Bearse et. al., 1997).

In general statistical modeling and model evaluation problems, the concept of model complexity plays an important role. At the philosophical level, complexity involves notions such as connectivity patterns and the interactions of model components. Without a measure of overall model complexity, prediction of model behavior and assessing model quality is difficult (Bozdogan & Bearse, 2003). Information theoretic ideas is a measure of overall model complexity in statistical modeling to help provide new approaches relevant to statistical inference.

Information theory is applied in a variety of fields and its abstract formulations as are applicable to any probabilistic or statistical system of observations. It plays an important role in modern communication theory, which formulates a communication system as a stochastic or random process (Kullback, 1997). Figure 4.1 shows many areas which are applied to information theory.

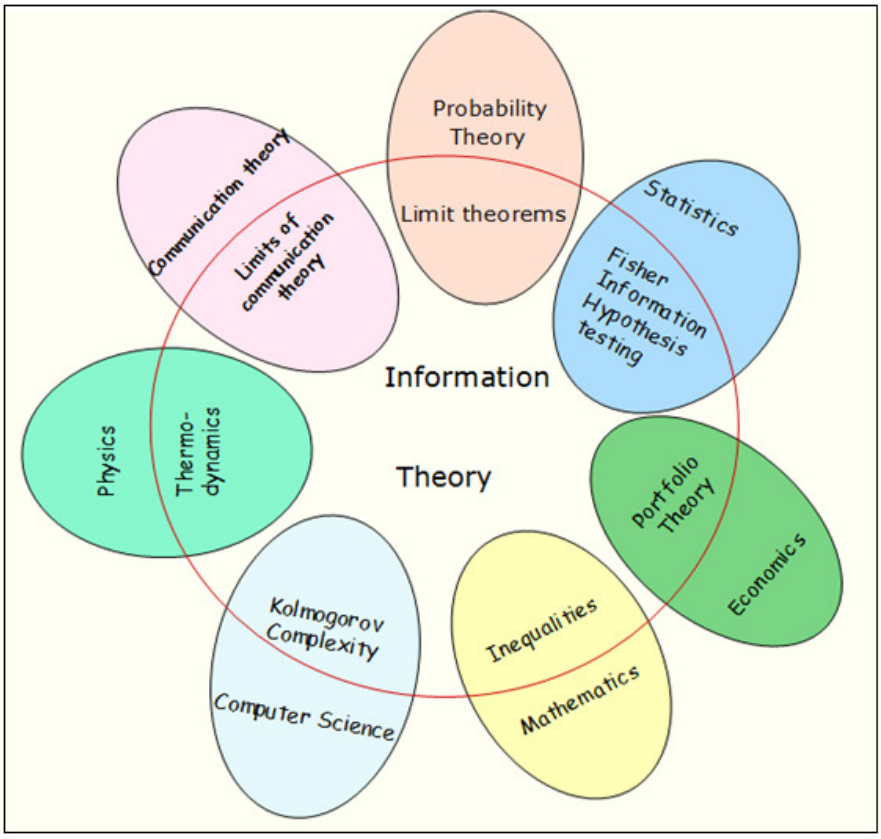

Figure 4.1 Areas which are applied to information theory (Kullback, 1997)

If a model should not be fitted to the data too closely; this is called over-fitting. Conversely, if the model is not well fitted to the data, it is said under-fitting. In both the cases, the model often becomes unstable and also has difficulty in prediction. Figure 4.2 shows a rough sketch of the relationship between model size - number of parameters and fitting error - prediction error for a generic information criterion.

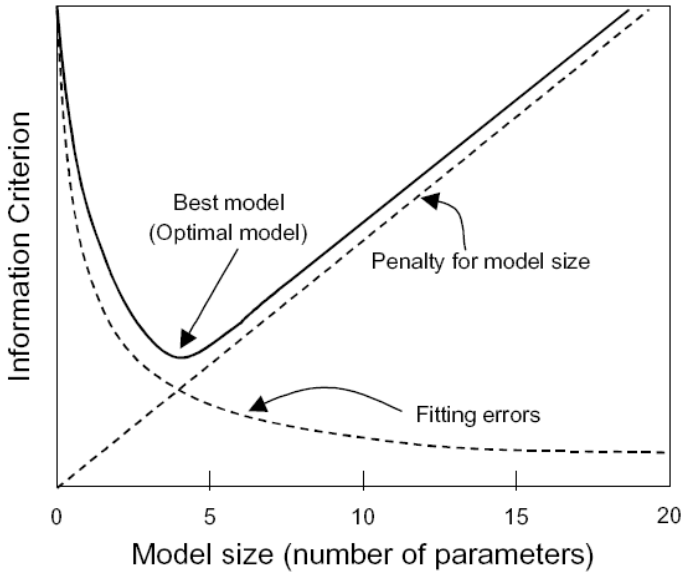

Figure 4.2 Information criterion as a function of model size

As the figure shows, information criteria are used to find the model that best balances model error against model size so as to prevent over-fitting or under-fitting of the data. It is considered that the minimum of the information criterion corresponds to the best optimal model size and the smaller the value, the better the model (Akaike, 1974; Judd, 2003).

4.1 Statistical Models to the Information Criterion

If the role of a statistical model is understood as being a tool for extracting information, it follows that a model is uniquely determined for a given object, but rather that it can assume a variety of forms depending on the viewpoint of the modeler and the available information. In other words, the purpose of statistical modeling is not to estimate or identify the unique or perfect model, but rather to construct a good model as a tool for extracting information according to the characteristics of the object and the purpose of the modeling. This means that, as a general rule, the results of inference and evaluation will vary according to the specific model. A good model will generally yield good results; however, one cannot expect to obtain good results when using an inappropriate model (Konishi & Kitagawa, 2008).

The general evaluation of the goodness of a statistical model is important to assess the closeness between the predictive distribution f(x) defined by the model and true distribution g(x), rather than simply minimizing models in terms of Kullback-Leibler information.

Estimation of Kullback-Leibler information is a crucial part of deriving a statistical model selection procedure which, like AIC, is based on the likelihood principle. In this section, it is given background information to allow a heuristic understanding of criteria used in selecting a model for making inferences from data. The first type of criteria AIC is estimates of Kullback-Leibler information or distance and attempt to select a good approximating model for inference, based on the principle of parsimony. The second type of criteria BIC is dimension consistent in that they attempt to consistently estimate the dimension of the true model. These latter criteria assume that a true model exists, that it is in the set of candidate models and that the goal of model selection is to find the true model, which in turn requires and that sample size is very large. The Kullback-Leibler based criteria do not assume a true model exists, let alone that it is in the set of models being considered. Based on a review of these criteria, also it is given ICOMP criteria.

4.2 Kullback-Leibler Information

Let $\mathbf{x_n} = \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}\}$ be a set of n observations drawn randomly and independently from an unknown probability distribution function G(x) which refer to the probability distribution function. It generates data as the true model or true distribution. In contrast, let F(x) be an arbitrarily specified model. If the probability distribution function G(x) and F(x) have density functions g(x) and f(x), respectively then they are called continuous models or continuous distribution models. If, given either a finite set or a countable infinite set of discrete points $\{x_1, x_2, ..., x_k, ...\}$, they are expressed as probabilities of events;

$$g_i = g(x_i) \equiv \Pr(\{\ \omega; X(\omega) = x_i\}),$$
$$f_i = f(x_i) \equiv \Pr(\{\ \omega; X(\omega) = x_i\}),\ i = 1,2,... \tag{4.1}$$

then these models are called discrete models or discrete distribution models. It is assume that the goodness of the model f(x) is assessed in terms of the closeness as a probability distribution to the true distribution g(x). As a measure of this closeness, it is used the following Kullback-Leibler information. It is abbreviated as K-L information (Konishi & Kitagawa, 2008):

$$I(G;F) = E_G\left[\log\left\{\frac{G(X)}{F(X)}\right\}\right], \tag{4.2}$$

where E_G represents the expectation with respect to the probability distribution G. K-L information can express for the discrete and continuous models as follows:

$$I(g;f) = \int \log\left\{\frac{g(x)}{f(x)}\right\} dG(x)$$
$$= \begin{cases} \int_{-\infty}^{\infty} \log\left\{\frac{g(x)}{f(x)}\right\} g(x)dx, & \text{for continuous model,} \\ \sum_{i=1}^{\infty} g(x_i)\log\left\{\frac{g(x_i)}{f(x_i)}\right\}, & \text{for discrete model.} \end{cases} \tag{4.3}$$

The K-L information has the following properties (Konishi & Kitagawa, 2008):

- **i.** $I(g;f) > 0$, whenever $g(x) \neq f(x)$,
- **ii.** $I(g;f) = 0 \Leftrightarrow g(x) = f(x)$.
- **iii.** If $(x_1, x_2, ..., x_n)$ are independent and identically distributed (i.i.d) random variables, then K-L information for the whole samples $I_n(g;f) = nI(g;f)$.

In view of these properties, it is considered that the smaller the quantity of K-L information, the closer the model f(x) is to g(x). The last property says that if the random variables are independent, K-L information is additive. K-L information can be used in actual modeling only in limited cases, since it contains the unknown distribution g, so that its value cannot be calculated directly. K-L information can be decomposed as following (Konishi & Kitagawa, 2008):

$$I(g,f) = E_G\left[\log\left\{\frac{g(x)}{f(x)}\right\}\right] = E_G\left[\log g(x) - E_G[\log f(x)]\right] \tag{4.4}$$

The first term on the right-hand side is a constant that depends solely on the true model g, it is clear that in order to compare different models, it is sufficient to consider only the second term right-hand side. This term is called the expected log-likelihood. The larger this value is for a model, the smaller its K-L information is and the better the model is. Since the expected log-likelihood can be expressed as (Konishi & Kitagawa, 2008).

$$\begin{aligned} E_G[\log f(X)] &= \int \log f(x)\,dG(x) \\ &= \begin{cases} \int_{-\infty}^{\infty} g(x)\log f(x)\,dx, \text{ for continuous models,} \\ \sum_{i=1}^{\infty} g(x_i)\log f(x_i), \text{ for discrete models,} \end{cases} \end{aligned} \tag{4.5}$$

it still depends on the true distribution g and is an unknown quantity that eludes explicit computation. However, if a good estimate of the expected log-likelihood can be obtained from the data, this estimate can be used as a criterion for comparing models. K-L information can express for the discrete and continuous models as follows (Konishi & Kitagawa, 2008):

$$I(g;f)=\begin{cases}\int_{-\infty}^{\infty}\log\left\{\dfrac{g(x)}{f(x)}\right\}g(x)dx, \text{ for continuous model,}\\ \\ \sum_{i=1}^{\infty}g(x_i)\log\left\{\dfrac{g(x_i)}{f(x_i)}\right\}, \text{ for discrete model.}\end{cases} \tag{4.6}$$

4.2.1 Bias Correction for the Log-Likelihood

In practical applications, it is difficult to precisely capture the true structure of given event a limited number of observed data. For this reason, it can be constructed several candidate statistical models based on the observe data at hand and select the model that most closely approximates the mechanism of the occurrence of the event. If it is considered the situation in which multiple models $\{f_j(z|\theta_j); j=1,2,...,m\}$ exist, and the maximum likelihood estimator $\hat{\theta}_j$ has been obtained for the parameters of the model θ_j. It appears that the goodness of the model specified by $\hat{\theta}_j$, that is, the goodness of the maximum likelihood model $f_j(z|\hat{\theta}_j)$, can be determined by comparing the magnitudes of the maximum log-likelihood $\ell_j(\hat{\theta}_j)$. However, it is known that this approach does not provide a fair comparison of models, since the quantity $\ell_j(\hat{\theta}_j)$ contains a bias as an estimator of the expected log-likelihood $nE_G[\log f_j(z|\hat{\theta}_j)]$, and the magnitude of the bias varies with the dimension of the parameter vector (Konishi & Kitagawa, 2008).

It is supposing that n observations x_n generated from the true distribution g(x) are realizations of the random variable $\mathbf{X_n}=(\mathbf{x_1},\mathbf{x_2},...,\mathbf{x_n})^{\mathbf{T}}$, and let,

$$\ell(\hat{\theta})=\sum_{\alpha=1}^{n}\log f(x_\alpha|\hat{\theta}(x_n))=\log f(x_n|\hat{\theta}(x_n)) \tag{4.7}$$

represent the log-likelihood of the statistical model $f(z|\hat{\theta}(x_n))$ estimated by the maximum likelihood method. The bias of the log-likelihood as an estimator of the expected log-likelihood from $E_G[\log f(z|\hat{\theta})] = \int \log f(z|\hat{\theta}) dG(z)$ is defined by

$$b(G) = E_{G(x_n)}[\log f(x_n|\hat{\theta}(x_n)) - nE_{G(z)}[\log f(z|\hat{\theta}(x_n))]] \tag{4.8}$$

where the expectation $E_{G(x_n)}$ is taken with respect to the joint distribution $\prod_{\alpha=1}^{n} G(x_\alpha) = G(x_n)$, of the sample x_n, and $E_{G(z)}$ is the expectation on the true distribution $G_{(z)}$. It is seen that the general form of the information criterion can be constructed by evaluating the bias and correcting for the bias of the log-likelihood as follows:

$$\begin{aligned} IC(X_n;\hat{G}) &= -2(\log \text{likelihood of statistical model - bias estimator}) \\ &= -2\sum_{\alpha=1}^{n} \log f(X_\alpha|\hat{\theta}) + 2\{\text{estimator for } b(G)\}. \end{aligned} \tag{4.9}$$

In general, the bias b(G) can take various form depending on the relationship between the true distribution generating the data and the specified model and on the method employed to construct a statistical model.

4.2.2 Estimation of Bias

The bias depends on the unknown probability distribution G that generated the data through $I(\theta_0)$ and $J(\theta_0)$. It must be estimated based on observed data. Let $\hat{I}$ and $\hat{J}$ be the consistent estimators of $I(\theta_0)$ and $J(\theta_0)$. In this case, it is obtained an estimator of the bias b(G) using,

$$\hat{b} = \text{tr}(\hat{I}\hat{J}^{-1}). \tag{4.10}$$

$I(\theta_0)$ and $J(\theta_0)$ matrices can be estimated by replacing the unknown probability distribution g(z) by empirical distribution function $\hat{g}_{(z)}$ based on the observed data as follows:

$$I(\hat{\theta}) = \frac{1}{n}\sum_{\alpha=1}^{n} \frac{\partial \log f(x_\alpha \mid \theta)}{\partial \theta} \frac{\partial \log f(x_\alpha \mid \theta)}{\partial \theta^T} \Big|_{\hat{\theta}} \tag{4.11}$$

$$J(\hat{\theta}) = -\frac{1}{n}\sum_{\alpha=1}^{n} \frac{\partial^2 \log f(x_\alpha \mid \theta)}{\partial \theta \partial \theta^T} \Big|_{\hat{\theta}} \tag{4.12}$$

4.3 Akaike Information Criterion (AIC)

The Akaike Information Criterion has played a significant role in solving problems in a wide variety of the fields as a model selection criterion for analyzing actual data. Akaike (1973) adopted Kullback-Leibler definition information. AIC is an asymptotically unbiased estimator of minus twice this expected log-likelihood. The AIC is defined by,

$$AIC = -2(\text{maximum log - likelihood}) + 2(\text{number of free parameters}) \tag{4.13}$$

The AIC is an evaluation criterion for the badness of the model whose parameters are estimated by the maximum likelihood method, and it indicates that the bias of the log-likelihood b(G) approximately becomes the number of free parameters contained in the model. The bias is derived under the assumption that the true distribution g(x) is contained in the specified parametric model $\{f(x \mid \theta); \theta \in \Theta \subset \Re^p\}$, that is, there exists a $\theta_0 \in \Theta$ such that the equality $g(x) = f(x \mid \theta_0)$ holds. The bias (4.9) of the log-likelihood is asymptotically given by,

$$E_G(x_n)\left[\sum_{\alpha=1}^{n} \log f(x_\alpha \mid \hat{\theta}) - nE_{G(z)} \log f(Z \mid \hat{\theta})\right] = tr\{I(\theta_0)J(\theta_0)^{-1}\} = tr(I_p) = p, \tag{4.14}$$

where I_p the identity matrix of dimension p. Hence, the AIC,

$$AIC = -2\sum_{\alpha=1}^{n} \log f(x_\alpha \mid \hat{\theta}) + 2p \tag{4.15}$$

can be obtained by correcting the asymptotic bias p of the log-likelihood.

The AIC does not require any analytical derivation of the bias correction terms for individual problems and does not depend on the unknown probability distribution G, which removes fluctuations due to the estimation of the bias.

The first term in equation (4.15) provides us with a measure of model inaccuracy badness of fit or lack of fit when the maximum likelihood estimators of the parameters in the model are used. The second term serves as a penalty for bias induced in the first term when additional free parameters are included in the model. Since the decision rule is to minimize AIC over the set of competing models, the best approximating model is chosen to be the one with the highest information gain (Peter et al., 1997).

There are several characteristics of AIC concerning the selection of a model using the AIC. These are explained in the following (Konishi & Kitagawa, 2008);

- The objective of modeling is to obtain a good model, rather than a true model. In situations where there are only a small number of observations, considering the instability of the parameters being estimated, the AIC reveals the possibility that a higher prediction accuracy can be obtained using models having lower orders.

- Shibata's (1976) described, if the true order is assumed, the asymptotic distribution of orders selected by the AIC can be a fixed distribution that is determined solely by the maximum order and the true order of family of models. This indicates that the AIC does not provide a consistent estimator of orders. However, that when the true order is finite, the distribution of orders that is selected does not vary when the numbers of observations is increased. In this case, even if a higher order is selected, when the number of observations is large, each coefficient estimate of a regressor with an order greater than the true order converges to the true value 0 and that a consistent estimator can be obtained as a model.

- Although the information criterion makes automatic model selection possible, the model evaluation criterion is a relative evaluation criterion. This means that selecting a model using an information criterion is only a selection from a family of models that have specified. Therefore, the critical task for researches is to up more appropriate models by making use of knowledge regarding that object.

4.4 Bayesian Information Criterion (BIC)

The Bayesian information criterion (BIC) or Schwarz's information criterion proposed by Schwarz (1978) is an evaluation criterion for models defined in terms of their posterior probability. AIC does not directly involve the sample size n and it has been criticized as lacking properties of consistency (Bozdogan, 1987). A popular alternative to AIC presented by Schwarz (1978) which does incorporate sample size is BIC.

Let $M_1,...,M_r$ be r candidate statistical models, and assume that each model M_i is characterized by a parametric distribution $f_i(x|\theta_i)$ $(\theta_i \in \Theta_i \subset \Re^{ki})$ and the prior distribution $\pi_i(\theta_i)$ of the k_i dimensional parameter vector θ_i. When n observations $\mathbf{X_n} = \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}\}$ are given, then, for the i^{th} model M_i, the marginal distributions or probability of x_n is given by,

$$p_i(x_n) = \int f_i(x_n \mid \theta_i)\pi_i(\theta_i)d\theta_i \tag{4.16}$$

This quantity can be considered as the likelihood of the i^{th} model and is referred to as the marginal likelihood of the data. According to Bayes' theorem, the posterior probability of the i^{th} model is given by,

$$P(M_i \mid x_n) = \frac{p_i(x_n)P(M_i)}{\sum_{j=2}^{r} p_j(x_n)P(M_j)},\ i = 1,2,..., \tag{4.17}$$

This posterior probability indicates the probability of the data being generated from i^{th} model when data x_n are observed. Therefore, if one model is to be selected from r models, it would be natural to adopt the model that has the largest posterior probability. This principle means that the model that maximizes the numerator $p_i(x_n)P(M_i)$ must be selected, since all models share the same denominator. If it is assumed that the prior probabilities $P(M_i)$ are equal in all models, it follows that the model that maximizes the marginal likelihood $p_i(x_n)$ of the data must be selected. Therefore, if an approximation to the marginal likelihood expressed in terms of an integral of $p_i(x_n)$ can readily be obtained, the need to compute the integral on a problem by problem basis will vanish, thus making the BIC suitable for uses a general model selection criterion. The BIC is actually defined as the natural logarithm of the integral multiplied by -2, and it is obtained (Konishi & Kitagawa, 2008),

$$\begin{aligned} -2\log p_i(x_n) &= -2\log\left\{\int f_i(x_n \mid \theta_i)\pi_i(\theta_i)d\theta_i\right\} \\ &\approx -2\log f_i(x_n \mid \hat{\theta}_i) + k_i \log n, \end{aligned} \tag{4.18}$$

where $\hat{\theta}_i$ is the maximum likelihood estimator of the k_i dimensional parameter vector θ_i of the model $f_i(x \mid \theta_i)$. Consequently, from the r models that are to be evaluated using the maximum likelihood method, the model that minimizes the value of BIC can be selected as the optimal model for the data. Thus, even under the assumption that all models have equal prior probabilities; the posterior probability obtained by using the information from the data servers to contrast the model and helps to identify the model that generated the data.

4.5 Information Complexity Criterion (ICOMP)

As an alternative to AIC, Bozdogan developed a new entropic statistical complexity criterion called ICOMP for model selection in general multivariate linear and non-linear structural models.

Motivated by considerations in AIC, Bozdogan's ICOMP criterion is based on the complexity of an element or set of random vectors via a generalization of van Emden's (1971) entropic covariance complexity index. Using an information theoretic interpretation, ICOMP views complexity as the discrimination information of the joint distribution of the parameter estimates against the product of their marginal distributions. Discrimination information is equal zero if the distributions are identical and is positive otherwise. The most general version of ICOMP is based on the estimated inverse Fisher information matrix (IFIM) of the model. For a general multivariate linear or nonlinear model defined by (Bozdogan & Bearse, 2003),

$$\text{Statistical model} = \text{signal} + \text{noise}. \tag{4.19}$$

ICOMP is designed to estimate a loss function:

$$\text{Loss Function} = \text{Lack of Fit} + \text{Lack of Parsimony} + \text{Profusion of Complexity} \tag{4.20}$$

in several ways. This is achieved by using the additivity properties of information theory. In the loss function, by the third term, profusion of complexity, it is mean that the interdependencies or the correlations among the parameter estimates and the random error term of a model (Bozdogan, 2000).

The development and construction of ICOMP is based on a generalization of the covariance complexity index instead of penalizing the number of free parameters directly, ICOMP penalizes the covariance complexity of the model. It is defined by (Bozdogan & Haughton, 1998),

$$\text{ICOMP} = -2\log L(\hat{\theta}_k) + 2C(\hat{\Sigma}_{\text{Model}}) \tag{4.21}$$

where $L(\hat{\theta}_k)$ is the maximized likelihood function, $\hat{\theta}_k$ is the maximum likelihood estimate of the parameter vector θ_k under the statistical model M_k, and C represents a

real-valued complexity measure and $\text{Cov}(\hat{\theta}) = \hat{\Sigma}_{\text{Model}}$ represents the estimated covariance matrix of the parameter vector of the model.

4.6 Information Criteria for Multiple Regression Models

In recent years, the statistical literature has placed more and more emphasis on model selection criteria. The problem is to choose the best approximating model among a class of competing models by a suitable model selection criterion given a finite data set. That model which optimizes the criterion is chosen to be the best model. In many situations where regression analysis is useful, the investigator has strong justification for including certain variables in the equation.

Model selection can be traced to back AIC criterion. Since then, a number of criteria have been proposed. While the AIC was proposed as an approximately unbiased estimator of the mean expected log-likelihood of a model, the BIC is an approximation to the posterior probability that a model is the best model. The ICOMP approximates the sum of two distances, one measuring the badness-of fit of the model, and one measuring its complexity (Bozdogan & Haughton, 1998).

AIC, BIC, and Information Complexity criterion are defined in multiple regressions as follows.

4.6.1 Akaike Information Criterion for Multiple Regression Models

The AIC is a frequently used method for regression and autoregressive model selection. In small samples, or when the number of fitted parameters is a moderate to large fraction of the sample size AIC can drastically underestimate the expected Kullback-Leibler information, and such bias can lead to severe overfitting. To remedy this shortcoming, it is proposed corrected of the AIC (AIC_c) statistic which is less biased and consequently tends to select much better models. For linear regression model selection AIC_c is an unbiased estimate for the Kullback-Leibler information. However,

AIC_c tends to overfit when the sample size increases (McQuarrie et al., 1997). These are explained as follows.

Suppose that the observations y_α are independently and normally distributed with mean μ_α and variance σ^2. Then the density function of y_α can be written as,

$$f(y_\alpha|\mu_\alpha,\sigma^2)=\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{(y_\alpha-\mu_\alpha)^2}{2\sigma^2}\right\} \tag{4.22}$$

by taking $\hat{\mu}_\alpha = x_\alpha^T\hat{\beta}$ $\hat{\sigma}^2=\frac{1}{(n-p)}\sum_{\alpha=1}^{n}(y_\alpha-x_\alpha^T\hat{\beta})^2$, then, the AIC for a Gaussian linear regression model is given by,

$$AIC_{(Regression)} = n\log(2\pi)+n\log(\hat{\sigma}^2)+n+2(p+1) \tag{4.23}$$

where p is the number of estimated parameters $(\hat{\beta}_0,\hat{\beta}_1,...,\hat{\beta}_p)$.

The bias corrected in AIC is approximated by the number of parameters which are constant and have no variability. Sugiura (1978) proposed corrected AIC as AIC_C in multiple regression model for small samples. AIC_C was used first by Hurvich and Tsai (1989). It is as follows,

$$AIC_C = n\log(2\pi)+n\log(\hat{\sigma}^2)+n+2(p+1)\frac{n}{n-p-2}. \tag{4.24}$$

AIC criterion may be viewed as an asymptotically unbiased estimator of the Kullback-Leibler information which is a measure of discrepancy between statistical models (Kullback & Leibler, 1951). Thus, selection of a model has minimizing AIC means that the selected model may be the best approximating model to the true model. For the data for which the true model has infinite order, AIC provides an asymptotically efficient selection of a finite order model. However, for the data for which the true

model has finite order, minimizing AIC does not produce consistent model order selection, which pursues the selection of the most parsimonious model. This defect is more evident when the sample size is very large. In other words, the existing asymptotically efficient criteria (e.g. AIC) which do not provide consistent order selection tend to overfit unless the maximum allowable order of the model is specified. This overfitting problem leads to more unsatisfactory model order selection when sample size is small, or when the number of free parameters is a relatively larger than the sample size. In this case, the overfitting stems from the fact that AIC is strongly negatively biased (Kwon, et al., 1998).

A key feature of an AIC type criterion is that it adds a penalty of the same order as the model dimension to the negative maximized log-likelihood. The significance of this is that with the penalty added as bias correction, the criterion value is of the same order as the sum of the squared bias and the estimation error. Consequently, when the number of the relevant models is under control, the comparison of the criterion value is pretty much similar to comparing the sum of the squared bias and the estimation error over the models. In light of the well known fact that the best trade-off between the squared bias and the estimation error typically produces the minimax optimal rate of convergence for both parametric and nonparametric function classes and, the AIC type criteria then have the property that they usually yield minimax-rate optimal estimators of the regression function under a squared error type loss. (Yang, 2003)

4.6.2 Bayesian Information Criterion for Multiple Regression Models

In multiple regression Schwarz (1978) Criterion is defined by,

$$\mathrm{BIC}_{(\mathrm{Regression})} = n\log(2\pi) + n\log(\hat{\sigma}^2) + n + p\log(n) \tag{4.25}$$

where n is the number of observations.

AIC(1973) and Schwarz (1978) are derived from distinct perspectives: AIC intends to minimize the Kullback-Leibler divergence between the true distribution and the

estimate from a candidate model and BIC tries to select a model that maximizes the posterior model probability. Due to the rather different motivations, it is not surprising that they have different properties (Yang, 2003).

The most well-known properties of AIC and BIC are asymptotic (loss) optimality and consistency (in selection), respectively. Simply put, when f is among the candidate families of regression functions, the probability of selecting the true model by BIC approaches 1 as $n \to \infty$. On the other hand, if f is not in any of the candidate families and if the number of models of the same dimension does not grow very fast in dimension, the average squared error of the selected model by AIC is asymptotically equivalent to the smallest possible offered by candidate models. These two properties of AIC and BIC are respectively called consistency and asymptotic optimally. Note that in general, AIC is not consistent and BIC is not asymptotically optimal in the nonparametric case (Yang, 2003).

4.6.3 ICOMP Based on Complexity Measures for Multiple Regression Models

In the literature many consistency results on AIC and BIC criteria are based on the central assumption that one of the models considered is true model. However, notably in the context of multiple regression, this assumption often does not hold, since one or more variables have been omitted from the model (Bozdogan & Haughton, 1998). Bozdogan and Haughton (1998), introduced a concept of consistency for this case, and established a consistency property for ICOMP criterion. Each formulation of ICOMP has the attractive feature of implicitly adjusting for the number of parameters, the sample size, and controlling the risks of both insufficient and over parameterized models. ICOMP inverse Fisher information matrix (ICOMP(IFIM)) is shown as for multiple regression [Bozdogan, 2000; Bozdogan, 2004).

$$\text{ICOMP(IFIM)}_{\text{Mul.Reg}} = n\log(2\pi) + n\log(\hat{\sigma}^2) + n + C_1(\hat{F}^{-1}(\hat{\theta})) \qquad (4.26)$$

Where

$$C_1(\hat{F}_R^{-1}(\hat{\theta})) = (p+1)\log\left[\frac{\mathrm{tr}(\hat{\sigma}^2(X'X)^{-1}) + \frac{2\hat{\sigma}^4}{n}}{(p+1)}\right] - \log\left|\hat{\sigma}^2(X'X)^{-1}\right| - \log\left(\frac{2\hat{\sigma}^4}{n}\right) \quad (4.27)$$

where $\hat{\sigma}^2$ is the estimated variance of regression model, and p explanatory variables in regression model. As the number of parameters increases (i.e., as the size of X increases), the error variance $\hat{\sigma}^2$ gets smaller even though the complexity gets larger. Also, as $\hat{\sigma}^2$ increases, $(X'X)^{-1}$ decreases. Therefore $C_1(\hat{F}^{-1}(\hat{\theta}))$ achieves a trade-off between these two extremes and guards against multicollinearity. To preserve scale invariance, it is used the correlational form of information fisher information matrix (IFIM), that is used $\hat{F}^{-1}$ and define the correlational form of ICOMP(IFIM) regression given by (4.26) (Bozdogan, 2004).

CHAPTER FIVE
INFORMATION CRITERIA METHOD TO DETECT OUTLIERS IN MULTIPLE REGRESSION USING GENETIC ALGORITHMS

Statistical models, particularly regression models, are most useful devices for extracting and understanding the essential features of datasets. However, most of the databases a particular amount of abnormal values, generally termed as outliers. An accurate identification of outliers plays a significant role in statistical analysis especially regression models. Nevertheless, many classical statistical models are blindly applied to data sets containing outliers; the results can be misleading at best. The appearance of outliers can exert negative influences on the fit of the multiple regression models.

This thesis integrates novel statistical modeling procedures based on an information theory. It is formed a three way hybrid between: the information criterion for outlier detection, multiple regression models, and genetic algorithm. It is demonstrated information criteria for outlier detection on a real and simulated data using GA. The format of this chapter is as follows:

- Information criterion for multiple regression models,
- GA for outlier detection,
- Real data examples and simulation studies for outlier detection in multiple regression models using genetic algorithms.

5.1 Detecting Outliers in Multiple Regression

An observation may have influence on estimates of the regression coefficients, the estimated variance of these estimates or the fitted values. The primary goal of the researcher should determine which influence to consider. After the detection of influential observation an even more task is the attempt to understand or explain the source of the influence. In this subsection begins with some background material on least squares regression estimation and diagnostics to identify a single outlier. The discussion expands to address the multiple outlier problems.

The study of outliers has interested practicing statisticians and other scientists for a great number of years. Thomson (1935) was the first author to drop both assumptions about population mean and standard deviation. Anscombe (1960) and Daniel (1960) were among the first authors to propose the use of standardized residual for detecting a single outlier in linear regression models.

Particular classes of diagnostic methods that are intended to aid in assessing the role those individual observations play in determining a fitted model. It seems that spurious observations may not always be outliers. It is therefore important for an analyst to be able to identify such observations, and assess their effect on various aspects of the analysis. Such observations called influential observations. A definition, which seems most appropriate, is given by Belsley et al., (1980): "An influential observation is the one which, either individually or together with several other observations, has a considerably larger impact on the calculated values of various estimates than is the case for the most other observations".

Isolating a single or a few outliers can be done quite easily using routine single-case diagnostics, that is, for example the Cook's squared distance (Cook, 1979) measures the change in the regression coefficients that would occur if a case was omitted. But the ordinary least squares (OLS) estimates and inference can be affected with the presence of even two outliers. However, the standard single-case diagnostic measures often suffer from masking and swamping. Masking is the inability of the procedure to detect the outliers, while swamping is the detection of clean observations as outliers.

5.2 Outlier Detection Methods in Multiple Regression

In regression type problems whether it is in multiple regression analysis, in logistic, or in ordinal logistic regression, model building and evaluation and selection of relevant subset of predictor variables on which to base inferences is a central problem in databases to reduce the "curse of dimensionality," a term coined by Richard Bellman (Bellman, 1961). Often a quantitative, binary, or ordinal level response variable is studied given a set of predictor variables. In such cases it is often desirable to determine

which subsets of the predictors are most useful for forecasting the response variable, and to interpret a large number of regression coefficients, since this can become unwieldy even for moderately sized data, better estimation and clearer interpretation of the parameters included in these models (Bozdogan, 2004).

The problem of selecting the best regression models is a significant exercise, particularly when a large number of predictor variables exist and the researcher lacks precise information about the exact relationships among the variables. In many cases the total possible number of models reaches over thousands (e.g., more than 20 predictor variables) and evaluation of all possible combinations of subsets is unrealistic in terms of time and cost (Bozdogan, 2004). Therefore, numerical optimization techniques and strategies for model selection are needed to explore the vast solution space. In general the problem of subset selection using numerical techniques requires two components: the efficient searching of the solution space, and a criterion or measure for the comparison of competing models to help guide the search.

Most statistical packages for statistical analysis provide a backward and forward selection strategy for choosing the best subset model. However, it is well known that both backward and forward stepwise selection in regression analysis do not always find the best subset of predictor variables from the set of p variables. Major criticisms leveled on backward and forward stepwise selection are that little or no theoretical justification exists for the order in which variables enter or exit the algorithm and the arbitrary choices of the probabilities specified a priori to enter and remove the variables in the analysis. Another criticism is that stepwise searching rarely finds the overall best model or even the best subset of a particular size. Lastly, and most importantly, because only local searching is employed, stepwise selection provides extremely limited sampling from a small area of the vast solution space. Stepwise selection, at the best, can only produce an adequate model (Bozdogan, 2004)..

Based on the above shortcomings of existing problems in regression analysis, in this chapter, it is introduced and developed a computationally feasible information criteria

based on the GA and information-based criteria for outlier detection in multiple regression models.

If outliers occur in the data, the errors can be thought to have a distribution different from the above. In multiple regression, outliers are typically modeled by either a shift in mean or via a shift in variance. It is adopted a variance inflation model for outliers in thesis. We assume that the errors with probability $(1-\pi)$ come from a $N(0,\sigma^2)$ distribution, and with probability π from $N(0,\varphi^2\sigma^2)$. Here π is the probability of an outlier and φ^2 is the variance inflation parameter.

The detection of outliers is an important issue for regression analysis, not only for their own sake, but also because the inferences drawn from the model will be biased if outliers are neglected. Belsley et al., (1980) described many of the well known outlier detection procedures for regression models (Belsley et al., 1980). Potential outliers can be incorporated into multiple regression models by the use of dummy variables in this study. A dummy variable is $n\times 1$ vector that has a value of one for the outlying observation, and zero for all other observations. Each outlier would have a corresponding a dummy variable like below data formulation.

$$\begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1p} & 1 & 0 & \dots & 0 \\ 1 & x_{21} & \dots & x_{2p} & 0 & 1 & \dots & 0 \\ . & . & . & . & . & . & . & \\ . & . & . & . & . & . & . & \\ . & . & . & . & . & . & . & \\ 1 & x_{n1} & \dots & x_{np} & 0 & 0 & \dots & 1 \end{bmatrix}$$

Figure 5.1 Regression models by the use of dummy variables

For example, we assume that the last observation is an outlier, then one dummy variable to be added to the model, and the explanatory variable matrix could be as below,

$$x_{n\times(p+1)} = \begin{bmatrix} x_{11} & \cdots & x_{1p} & 0 \\ . & \cdots & . & . \\ . & \cdots & . & . \\ . & \cdots & . & . \\ x_{n1} & \cdots & x_{np} & 1 \end{bmatrix}$$

Figure 5.2 The explanatory variable matrix with one dummy variable

A dummy variable in the regression model is therefore equivalent to detected outlier, and the problem here is the selection of the best model, where the candidate models have different combinations of all possible dummy variables as explanatory variables.

5.3 Information Criteria for Outlier Detection

The outlier detection problem in multiple regression can be viewed as two issues: The first is to define which data can be considered as inconsistent or exceptional in a given data set, and the second find an efficient method to detection of outliers. Based on these issues, in this part of the thesis, it is introduced detection of outliers is based on the use of information criteria in multiple regression models. Numerous methods have been proposed in the literature for outlier detection using AIC or its variants in linear regression.

Kitagawa & Akaike (1982) and Kitagawa (1984) have applied AIC in detection of outliers by using quasi Bayesian approach with predictive likelihood in place of the usual likelihood function.

Outlier detection is a statistical problem that has received considerable attention, both from the Bayesian and frequents perspectives. A common approach consists in assuming that the possible outliers are generated by contaminating models different from the one generating the rest of the data. In the usual formulation, data is assumed to be a random sample $x_1, x_2, ..., x_n$ from a model $f(x|\theta)$. In the outlier detection scenario, some few specific observations are suspected to be outliers, that is, to be generated by some other

contaminating models h_i. If the model h_i are specified, the result is another possible model for all the data. In this way, outlier detection can be reduced to a problem of model selection and the appropriate Bayesian procedures could then be used (Bayarri & Morales, 2003).

Bayesian approach for outlier detection in multivariate samples is also proposed by Guttman (1973). The approach assumes that the underlying distribution is $N(\mu,\Sigma)$ and there is one observation which comes from $N(\mu+\alpha,\Sigma)$. Guttman suggests using the posterior distribution of α to detect outliers. Another Bayesian approach was proposed by Varbanov (1998). This approach uses the posterior distribution of the squared norm of the error terms to detect outliers in the multivariate linear models. Other contributions to detection of multivariate outliers are included in Gnanadesikan & Kettering (1972), Hawkins (1980), Rousseuw & Leroy (1987), Varbanov (1998). Ting et al., (2007) introduced a Bayesian way of dealing with outlier infested sensory data and developed a block box approach to the removed outliers in real time.

The model selected by BIC is consistent with the true model asymptotically. It is known that BIC tends to underestimate the size of parameters in general; however, the properties in small samples are not well checked (Shono, 2000).

Bozdogan (2003) modified the ICOMP criteria for masking issues within the vector autoregression (VAR) context.

5.4 Adapting Information Criteria to Outlier Detection by Adding Penalty Terms

All the outlier detection effort in this study is based on the use of information criteria. There is large number of possible information criteria to choose from. The AIC, BIC and ICOMP will be used here, since they usually perform quite well in various situations. These methods mentioned above show the local influence, and they can not be resistant to masking and swamping effects as discussed in Suárez Rancel & González Sierra (1999). Therefore the information criteria can be modifying in order to handle the outliers. Bozdogan (2003) modified the ICOMP criteria for masking issues within the

vector autoregression (VAR) context. Tolvi (2004) used corrected BIC for outlier detection in multiple regression.

For multiple regression models with dummy variables the BIC criteria can be calculated as,

$$BIC = \log(\hat{\sigma}^2) + m\log(n)/n \tag{5.1}$$

where $\hat{\sigma}^2$ is the estimated variance of regression model. The total number of parameters in the estimated model are $m = 1 + p + m_d$,which consists of parameters for the constant, p is number of independent variables, and m_d is number of outlier dummies in regression model. In the equation, n is the number of observations. Generally a good model has small residuals, and few parameters, then it is preferred chosen with the smallest value of BIC.

A problem in using the BIC for outlier detection is that it tends to include unnecessary outlier dummies by itself. To circumvent this problem, a correction to the criterion is used. The corrected BIC takes into account the different nature of outlier dummies and other variables, and has a different penalty term for different variables. This takes the form of an extra penalty for the dummies. The corrected BIC is denoted BIC', and it is given by Tolvi (2004),

$$BIC' = \log(\hat{\sigma}^2) + (1+p)\log(n)/n + \kappa m_d \log(n)/n, \tag{5.2}$$

where the Kappa $(\kappa > 1)$ is the extra penalty given to outlier dummies, and m_d is number of dummy variables in regression model for each outlier observation.

In this thesis, we derive an AIC' and ICOMP' criteria alternative to BIC' approach for outlier detection in multiple regression. Our objective is to provide a more judicious penalty term than BIC', since counting and penalizing the number of parameters for outlier dummies in a model is necessary.

We generate and investigate penalty terms versions of AIC and ICOMP for outlier detection in multiple regression models, they are denoted AIC' and ICOMP'. The penalty term for the dummies in AIC' and ICOMP' criteria is defined as $\kappa m_d \log(n)$. They are given as follows, respectively.

AIC' for outlier detection in multiple regression models is given by,

$$\begin{aligned} AIC' &= AIC + \kappa m_d \log(n) \\ &= n\log(2\pi) + n\log(\hat{\sigma}^2) + n + 2(p+1) + \kappa m_d \log(n) \end{aligned} \tag{5.3}$$

Also, ICOMP' for outlier detection in multiple regression models defined by,

$$\begin{aligned} ICOMP(IFIM)'_{Mul.Reg} &= ICOMP(IFIM) + \kappa m_d \log(n) \\ &= n\log(2\pi) + n\log(\hat{\sigma}^2) + n + C_1(\hat{F}^{-1}(\hat{\theta})) + \kappa m_d \log(n) \end{aligned} \tag{5.4}$$

where,

$$C_1(\hat{F}_R^{-1}(\hat{\theta})) = (p+1)\log\left[\frac{tr(\hat{\sigma}^2(X'X)^{-1}) + \frac{2\hat{\sigma}^4}{n}}{(p+1)}\right] - \log\left|\hat{\sigma}^2(X'X)^{-1}\right| - \log\left(\frac{2\hat{\sigma}^4}{n}\right) \tag{5.5}$$

We illustrate the practical utility and the importance of this new outlier detection criterion by providing simulation examples for comparing their performance against BIC'. Also, simulation experiments are conducted to determine relevant different values of Kappa (κ) for outlier detection.

5.5 Genetic Algorithms Based Outlier Detection

Heuristic search methods such as genetic algorithms, swarm intelligence, ant colony optimization algorithms, tabu search, simulated annealing have been used to solve

various problems and produce good solutions with probably good run times. The most widely used form of evolutionary computation is the genetic algorithm. GAs have proven to be a very successful meta-heuristic technique for many NP-complete optimization problems (Chen et al., 2007; Maaranen, 2007; Toroslu & Arslanoglu, 2007). GAs are stochastic search methods that have been successfully applied in many search, timetabling, scheduling, and machine learning problems and have been especially used in engineering, biology, and medicine (Uğur, 2008). GAs are inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (Goldberg, 1989). Note that, the outline of GA is given in Figure 2.5.

Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified to form a new population. The new population is then used in the next iteration of the algorithm.

GAs are useful optimization tool for statistical modeling, also it has been used for outlier detection and model selection of linear regression models or times series. Crawford and Wainwright (1995) presented genetic algorithms capable of generating subsets for multiple-case outlier diagnostics. The genetic algorithms are used the diagnostics as evaluation functions to drive the search for good subsets. Jann (2000) describes GAs for the detection of level shifts in a time series. Ishibuchi et al., (2001) used GAs for the feature selection in data mining and they gave a lot of references about this literature. Additionally, the use of GAs for outlier detection and variable selection is proposed by Tolvi (2004).

In this subsection, a GA has been applied to detection of outliers in multiple regression as an optimization search algorithm. An important issue in applying genetic algorithms is to represent a candidate solution as a chromosome. Therefore, it is given chromosome representation method and properties of genetic algorithm operators for outlier detection. The GA for outlier detection is written using Matlab environment and

GA codes are given appendice 1. The primary elements of GAs for outlier detection are shown as follows in details.

• Parameter Encoding and Chromosome Structure: The coding of the candidate solutions for outlier detection is straightforward. Each solution also called an individual or chromosome, is fully described by a binary vector d, it is defined as $d = (d_1,...,d_i,...,d_n)$, where $d_i = 0$ indicates no outlier dummy and $d_i = 1$ indicates an outlier dummy for observation i, for each $i = 1,...,n$. A dummy variable for outlier observation must be created during the GA is run on a data set.

The structure of a chromosome in GAs is shown in Figure 5.3 for this study. It has n genes which is the number of observations in the data set. Each chromosome consists of p genes, where p is the number of outliers given in a model. For instance, the second and (n-1)th observations are outliers and d vector is defined as $d = (010...010)$.

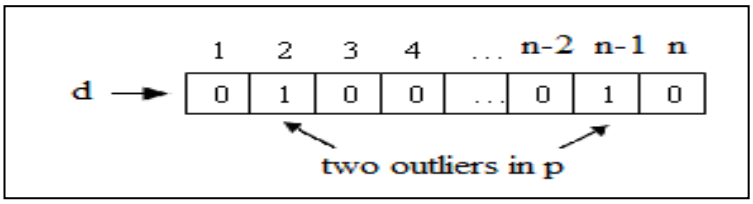

Figure 5.3. The Structure of a chromosome in genetic algorithm

• The Population and Generations: The population size is the number of chromosomes in each generation. Initial population is usually generated randomly. Multiple individuals are selected from current population and used to form new generations. The individuals with smallest values of the fitness function are more likely to pass their genes onto the next generation.

Multiple regression models corresponding to these individuals are then estimated by using the observed data, and information criterion values for them are computed.

• Fitness Function: Fitness function is a function that quantifies of a solution chromosome or individual in a genetic algorithm. A new population is obtained through selecting the most promising individuals. Since the individuals with a high fitness value

are more frequently selected, there is pressure for the fitter individuals to be incorporated into the population.

The fitness of an individual is computed by one of the information criterion ($AIC', BIC', ICOMP'(IFIM)$) for multiple regression models with the corresponding dummy variables in this study.

• Selection Operator: The selection strategy is largely dependent upon the fitness level of the individuals actually existing in the population. There are various selection strategies based on fitness, the most commonly used one of which is the fitness proportion selection. Tournament selection and ranking selection are the other two alternative strategies.

Stochastic uniform selection function is used for GA in this study. This function lays out a line in which each parent corresponds to a section of the line of length proportional to its scaled fitness value. The algorithm moves along the line in steps of equal size. At each step, algorithm allocates a parent from the section it lands on. It is noted that the results can be improved if a small number of the best individuals. These are kept the same from one generation to the next. The best two individuals are kept as elite population.

• Crossover Operator: Crossover is a genetic operator that combines (mates) two chromosomes (parents) to produce a new chromosome (offspring). This operator is seen Figure 5.4, where the "|" symbol indicates the randomly chosen crossover point.

Parent 1	+	Parent 2	=	Offspring1	Offspring2
11001|010	+	00100|111	=	11001|111	00100|010

Figure 5.4 Crossover operator

The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. Crossover is performed during evolution according to a user-defined crossover probability with P_C.

This procedure is repeated until the same numbers of individuals are constructed in the previous generation.

- Mutation Operator: Mating of the individuals from the previous generation will not be enough for diversity of population. In evolutionary terms, small changes in chromosomes are needed for genetic variations. To this end, the individuals of each generation are also mutated before model estimation. Each gene of each individual is flipped, from zero to one or vice versa with mutation probability P_m.

Table 5.1 Summarize the parameters of GA for the simulated models in this study.

Table1. The Parameters of the GA for the Simulated Model

Number of Generations	250
Population Size	40
Fitness Value	$AIC', BIC', ICOMP'(IFIM)$
Selection Operator	Stochastic uniform selection
Crossover Probability	1
Mutation Probability	0.01
Elitism	For two parents

In addition to crossover and mutation, a condition for the maximum number of dummies is used to alter the population. Since only a few dummies will be allowed in the final model, this condition is used in order to keep the candidate models from having too many variables. The rule states that if a candidate model has more than n/2 dummy variables, or outliers is more than 50% of the number of observations, it is ignored consideration. Depending on the particular crossover and mutation rates P_C and P_m, the next generation will be composed entirely of offspring models or of a mixture of offspring and parent models.

5.6 Design of Simulation Study and Experimental Results

To gain a better understanding of $AIC', BIC', ICOMP'(IFIM)$ criterion performance for the outlier detection problem with multiple outliers using GAs, it is run a designed experiment using Monte Carlo simulation. The experiment has varies characteristics not only of the data set, but also the Kappa value of information criteria in order to quantify

the expected performance of information criterion. In the next subsection, it is showed that real data examples, the steps of data generation and simulation results of new approaches to detect outliers by means of a simulation study.

5.6.1 Real Data Examples for Outlier Detection using Genetic Algorithms

Two experimental data sets have been used to illustrate outlier detection in MLR modeling. References to these, and other information, including where to obtain the data can be found in (Hoeting et. al., 1996)[‡]. In this subsection, it is investigated that detect outliers from these data sets with GA. Some information on the data sets and results are following;

i Scottish Hill Racing: The first example involves data supplied by Scottish Hill Runners Association (Hoeting et al., 1996). The purpose of the study is to investigate the relationship between record time of 35 hill races and two explanatory variables: distance is the total length of the race, measured in feet. One would expect that longer races and larger climbs would be associated with longer record times.

Several authors have examined these data sets using both predictors in their analysis (Atkinson, A.C., 1986; Hadi, 1986; Hoeting et .al., 1996). They concluded that races 7^{th} and 18^{th} observations are outliers. After they removed observations 7 and 18, their methods indicated that observation 33 is also an outlier. Thus observations 7 and 18 mask observation 33. After race numbers 7,18, and 33 are removed from the data, standard diagnostic checking does not reveal any gross violations of the assumptions underlying MLR models (Fox, 1997; Hoaglin & Tukey, 1983; Hoeting et al., 1996). The scatter plot of this data set is shown in Figure 5.5.

[‡] These data sets are available from one of the authors' website. This web address is http://www.stat.colostate.edu/~jah/index.html

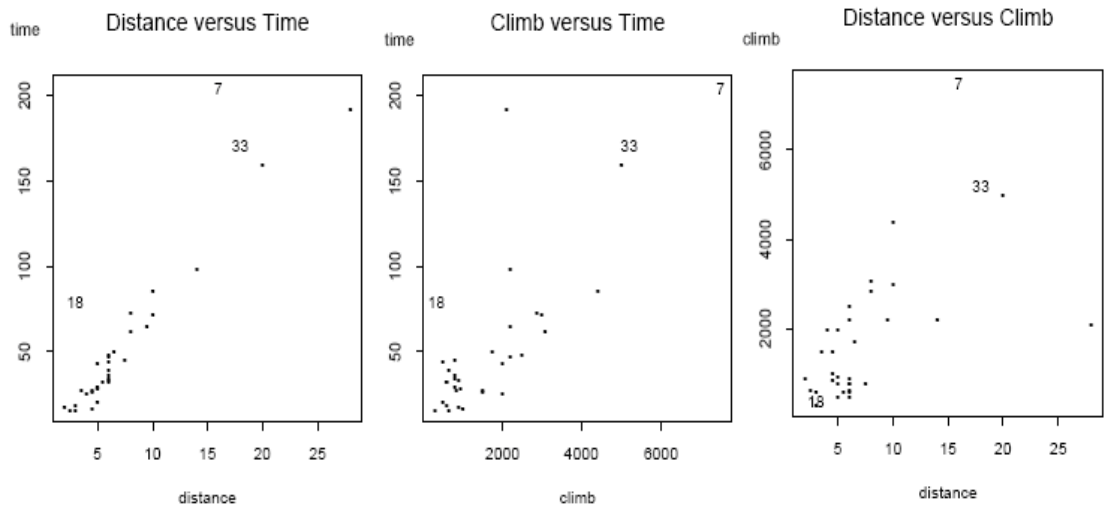

Figure 5.5 Scatter Plot of Scottish Hill Racing Data.[§]

The GA described earlier was run many times with this data; all runs result in the same outliers being detected, at observations 7,18, and 33. The solution was always found quickly by the GA and it is seen Figure 5.6. There are four windows in Figure 5.6. The first window showed that the vector entries of the individual with the best fitness function's value in each generation. It is found the best individuals 7, 18, 33. The second window showed that the best function value in each generation versus iteration number. Then, the optimal fitness function value of GA has a BIC' value 4.13. The third window is showed that the fitness of each individual in each iteration. The last window is showed that stopping criteria levels. It is specifies the maximum number of iterations the genetic algorithm performs. It is selected 100 in GA for this example.

[§] Numbers correspond to Race Numbers 7, 18, 33. Distance is given in miles, time is given in minutes, and climb is given in feet.

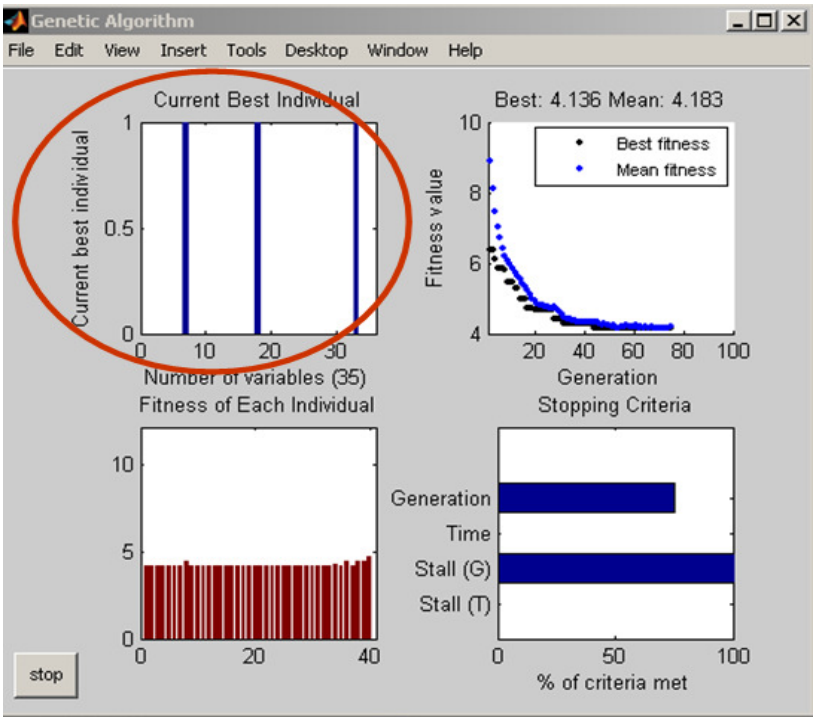

Figure 5.6 The output of genetic algorithm for the Scottish Hill Racing data

ii The Stack Loss Data: The stack loss data consist of 21 days of operation form a plant for the oxidation of ammonia as a stage in the production of nitric acid. The response is called stack loss which is the percent of unconverted ammonia that escapes from the plant. There are three explanatory variables. The air flow is first independent variable which measures the rate of operation of the plant. The second independent variable measures the inlet temperature of cooling water circulating through coils in this tower and the last independent variable is proportional to the concentration of acid in the tower. Small values of the respond correspond to efficient absorption of the nitric oxides. In earlier research (Atkinson, 1986; Hoeting et. al., 1996) been identified as outliers four observations. These are 1,3,4, and 21 observations. This data set provides an interesting extreme example of masking (Atkinson, 1986). The detection of any of these outliers is very difficult if only one observation at a time is examined. But the simultaneous methods are able to detect all of four outliers at a time.

The GA was run a lot of times with this data. The entire run gives to result in the same outliers being detected, at observations 1,3,4, and 21. The best outlier combination

was always found quickly by the GA as seen in Figure 5.7. The optimal fitness function value of GA has a BIC′ value 2.29.

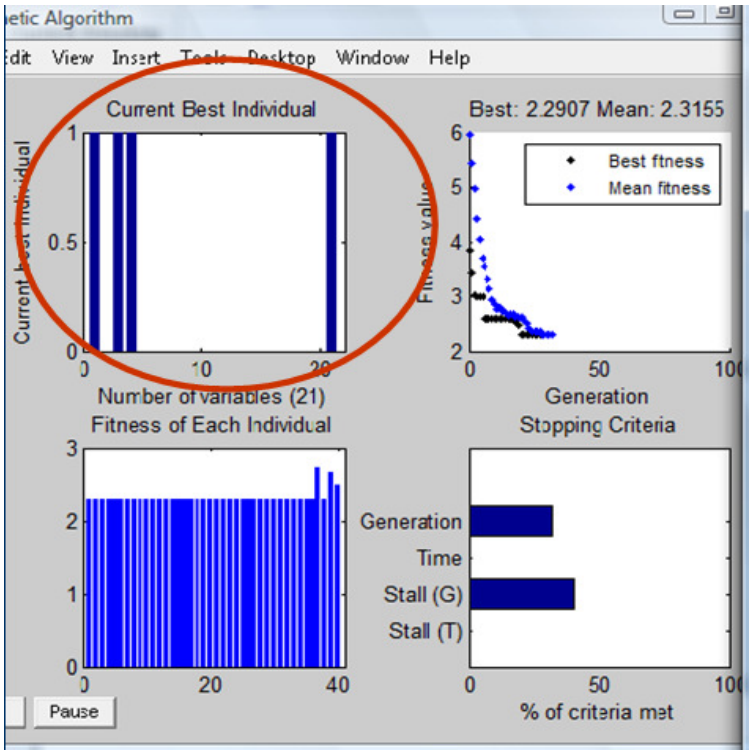

Figure 5.7 The output of genetic algorithm for the Stack Loss data

5.6.2 Generating Simulated Data Sets

In this subsection, it is performed Monte Carlo experiments to evaluate the performance comparisons of new approaches information criterion. To carry out simulations run, it is preceded on different regression models, these are explained as follows.

The response vector is generated as $Y = X\beta + \varepsilon$ where X is the design matrix of variables from the normal distribution. β is the vector of known parameters and e_i is the vector of random errors from a standard normal distribution. A detailed description of the regression models can be summarized in Table 5.2.

Table 5.2 Multiple regression models and varieties

Multiple Regression Models	X_i	Errors
$y_i = 0.10 + 1.08x_{1i} + 1.48x_{2i} + e_i$	$X_1 \sim N(0,1)$	$e_i \sim N(0,1)$
$y_i = 0.10 + 1.06x_{1i} + 1.38x_{2i} + 1.32x_{3i} + e_i$	$X_2 \sim N(2,1)$	
$y_i = 0.10 + 1.21x_{1i} + 1.34x_{2i} + 1.10x_{3i} + 1.88x_{4i} + e_i$	$X_3 \sim N(3,1)$	
$y_i = 0.10 + 1.60x_{1i} + 1.15x_{2i} + 1.33x_{3i} + 1.16x_{4i} + 1.73x_{5i} + e_i$	$X_4 \sim N(2.3,1)$	
	$X_5 \sim N(2.5,1)$	

Then n regression observations were generated randomly according to the models in Table 5.2. For every set of n observations that have been used in this research. This experiment for simulated data sets include 4 factors, these are chosen as below,

Factors	Levels
Number of regressor variables (p)	2, 3, 4, 5
Number of observations in data set (n)	20, 30, 40, 50, 60, 70, 80, 90, 100
The percentage of outliers in y_i (ω)	5, 10
The penalized value (κ)	3, 4, 5

These can be defined as n: number of observations, p: number of explanatory variables in regression model, ω: percentage of outliers contaminating the sample. The outlier density levels are selected as 5% and 10%, and κ is penalized value of information criteria.

Firstly, the response variable y_i is generated for each of sample sizes. After the y_i generated from normal distribution, the outliers are generated from the uniform distribution which lie at least +3 standard deviations from the mean of Y taken into account of percentage of outliers. The number of outliers must be added in Y for each sample sizes and the percentages of outliers are given in Table 5.3.

Table 5.3 The Number of Outliers must be added in Y for each Sample Sizes and the Percentage of Outliers

	n								
ω	20	30	40	50	60	70	80	90	100
5	1	2	2	3	3	4	4	5	5
10	2	3	4	5	6	7	8	9	10

For each of the combinations of parameters in Table 5.3 are generated 100 data sets take into account dimension of regression models, so, it is generated a total of 7200 data sets. Then, data sets are applied to AIC′, BIC′, and ICOMP′(IFIM) information criterion with their penalized values for Kappa=3, 4, and 5.

5.6.3 Comparison of Performances of Some Information Criteria for Outlier Detection using Genetic Algorithm

In this subsection, the GA is used for outlier detection and variable selection in multiple regression model and information criteria are used as the fitness function of GA. The GA is proceeded to find the optimal solution through for each combination of experiments. Each dataset containing known percentage of outliers and the genetic algorithm was exceptionally detecting these outliers in all of the dataset tested. The best models have been chosen according to value of information criteria from most of the generations by GA, and GA can detect the outliers simultaneously search in the solution space, therefore GA based outlier detection method allows for detection of multiple outliers, not just one at a time.

Performance of AIC′, and ICOMP′(IFIM) information criteria are compared against through simulation experiments. The value of information based selection criterion is calculated for observation as a measure of the fitness of dependent variable in multiple regression models and also it is used in different penalized value Kappa (κ) for each information criterion.

The simulation results are reported in Table 5.4 to Table 5.7 and the values are the percentage of outliers (P_{out}) for 100 replicates. The P_{out} score tested the performance of new approaches of information criterion. It tests the performance of information

criterion under two components. These are numbers of incorrectly identified observations as outliers (I_{out}) and numbers of failure to identify any of the outliers (F_{out}) in all iterations for each subsets. P_{out} is calculated by,

$$P_{out} = \frac{I_{out} + F_{out}}{T_{out}} \times 100\% \tag{5.6}$$

where T_{out} is total numbers of outliers for all iterations in each subsets. Percentages of outliers are given in Table 5.4 to Table 5.7 and these are obtained by 100 replications for all contaminating samples and levels of Kappa values, in which penalized value tend to increase performance of information criterion for outlier detection.

The results of simulations for percentage of outliers 5% in Y against all sample sizes and all dimensions are given in Table 5.4 and Table 5.5. It illustrates the ability of the information criterion in detecting outliers for all situations. For n=20 and κ<5 both the AIC′, and ICOMP′(IFIM) criteria perform better than the BIC′ and, also ICOMP′(IFIM) criteria outperforms BIC′. When $n \geq 30$ the tendency to overfit for all criteria decreases slightly, but ICOMP′(IFIM) provides the best detection of outliers.

The results of simulations for percentage of outliers 10% in Y against sample sizes and all dimensions are seen in Table 5.6 and Table 5.7. When n>30, a tendency to overfit for all criteria decreases slightly against dimension of regression models, but ICOMP′(IFIM) provides the best detection of outliers in regardless of dimensions of regression model.

Tablo 5.4 P_{out} for 5% Outlier in Y using GA by Fitness Function . AIC′, BIC′, ICOMP′(IFIM)

		p=2			p=3		
		Kappa Values			Kappa Values		
n	Information C.	3	4	5	3	4	5
20	AIC'	3.00	1.00	0.00	10.00	28.00	24.00
	BIC'	2.00	5.00	1.00	131.00	58.00	24.00
	ICOMP'	0.00	0.00	0.00	1.00	0.00	0.00
30	AIC'	0.00	0.00	0.50	0.50	0.00	0.00
	BIC'	3.00	0.00	0.00	17.50	1.50	0.00
	ICOMP'	0.00	0.50	0.00	2.00	0.00	0.00
40	AIC'	2.00	0.50	0.00	1.00	1.50	1.00
	BIC'	5.50	1.50	0.50	8.00	2.00	1.00
	ICOMP'	1.50	0.50	0.00	2.50	1.00	1.50
50	AIC'	0.67	0.33	1.00	3.00	2.33	1.00
	BIC'	1.33	0.33	0.33	6.33	4.00	2.67
	ICOMP'	0.67	0.67	1.00	3.33	1.67	2.67
60	AIC'	2.67	2.00	1.00	3.67	2.33	3.33
	BIC'	1.00	1.00	1.00	7.33	1.67	3.67
	ICOMP'	2.00	2.33	0.67	3.33	2.33	2.00
70	AIC'	3.50	2.75	3.25	4.75	4.25	2.50
	BIC'	2.50	2.00	2.00	7.00	4.25	2.25
	ICOMP'	1.50	3.25	2.75	5.50	4.00	3.25
80	AIC'	3.75	4.25	5.00	6.00	7.25	5.50
	BIC'	4.25	4.00	4.00	11.25	7.00	7.00
	ICOMP'	6.25	3.25	4.00	6.75	6.25	4.75
90	AIC'	6.20	10.80	9.00	6.80	8.00	6.00
	BIC'	15.60	6.80	7.80	10.40	7.80	5.20
	ICOMP'	6.00	7.80	7.60	11.00	7.40	7.00
100	AIC'	12.00	11.40	9.00	15.00	11.80	9.00
	BIC'	9.80	9.80	10.00	17.60	12.20	11.20
	ICOMP'	10.00	8.20	9.20	11.40	9.60	8.20

Tablo 5.5 P_{out} for 5% Outlier in Y using GA by Fitness Function . AIC′, BIC′, ICOMP′(IFIM)

		p=4			p=5		
		Kappa Values			Kappa Values		
n	Information C.	3	4	5	3	4	5
20	AIC'	332.00	193.00	110.00	519.00	360.00	294.0
	BIC'	341.00	160.00	71.00	601.00	413.00	232.0
	ICOMP'	2.00	1.00	0.00	3.00	2.00	2.00
30	AIC'	36.00	1.00	1.00	40.00	48.50	1.50
	BIC'	17.50	6.00	4.00	54.50	3.00	5.50
	ICOMP'	1.00	0.50	0.00	0.50	0.00	0.50
40	AIC'	1.00	1.50	0.00	15.50	4.50	0.00
	BIC'	8.00	2.00	1.00	18.00	3.00	1.50
	ICOMP'	1.50	2.00	0.00	1.00	0.00	0.50
50	AIC'	1.00	0.00	0.67	7.67	3.67	1.33
	BIC'	3.67	1.00	2.00	7.33	3.00	2.67
	ICOMP'	1.33	0.67	0.67	3.67	2.33	1.00
60	AIC'	3.33	2.67	2.00	5.67	3.33	1.67
	BIC'	8.00	3.67	3.00	5.67	3.00	2.33
	ICOMP'	4.00	2.67	3.00	3.00	4.00	1.67
70	AIC'	6.50	35.50	3.75	5.50	3.25	3.25
	BIC'	7.00	4.25	3.50	6.00	3.75	4.25
	ICOMP'	6.50	4.00	3.25	3.25	3.75	4.75
80	AIC'	5.25	5.75	6.00	6.75	5.00	4.25
	BIC'	8.00	8.25	5.25	6.75	5.75	6.75
	ICOMP'	6.00	4.75	4.25	5.75	4.50	3.50
90	AIC'	8.40	8.80	6.00	10.20	8.20	7.00
	BIC'	10.20	7.40	8.00	8.80	7.40	7.40
	ICOMP'	9.20	6.20	6.60	7.20	8.20	5.60
100	AIC'	21.80	19.00	15.60	13.00	11.40	10.00
	BIC'	22.80	18.80	16.00	10.40	11.80	10.20
	ICOMP'	19.60	16.60	15.40	10.00	12.20	9.60

Tablo 5.6 P_{out} for 10% Outlier in Y using GA by Fitness Function AIC′, BIC′, ICOMP′(IFIM).

		p=2			p=3		
		Kappa Value			Kappa Value		
n	Information C.	3	4	5	3	4	5
20	AIC'	3.50	0.00	0.00	20.50	12.00	2.00
	BIC'	23.50	0.50	0.00	112.50	79.00	8.00
	ICOMP'	0.00	0.00	0.00	1.50	0.00	0.00
30	AIC'	0.00	0.33	0.67	1.00	0.00	0.00
	BIC'	0.67	0.00	0.00	11.67	1.00	1.00
	ICOMP'	0.00	0.00	0.33	1.33	0.00	0.00
40	AIC'	4.00	1.00	1.25	8.50	1.00	1.50
	BIC'	4.00	1.00	1.25	8.50	1.00	1.50
	ICOMP'	1.00	0.25	2.50	0.50	0.50	0.50
50	AIC'	0.80	15.0	16.80	1.80	2.00	12.80
	BIC'	0.80	5.80	3.60	5.40	5.00	17.60
	ICOMP'	0.80	0.60	4.80	2.00	1.20	13.00
60	AIC'	1.50	14.0	19.33	8.00	17.83	22.83
	BIC'	0.67	3.00	7.83	4.33	3.67	10.67
	ICOMP'	0.83	1.17	16.17	1.50	0.83	20.17
70	AIC'	8.57	11.71	37.14	16.43	31.71	36.00
	BIC'	2.43	8.57	20.29	5.14	14.57	24.43
	ICOMP'	1.00	1.43	27.86	3.29	1.29	24.57
80	AIC'	11.25	28.63	52.63	23.50	40.13	70.38
	BIC'	5.00	18.50	43.50	5.13	14.50	38.63
	ICOMP'	1.75	3.88	45.75	2.75	1.88	52.63
90	AIC'	13.44	57.4	72.67	67.44	56.78	72.33
	BIC'	2.33	30.0	61.22	5.67	25.56	68.11
	ICOMP'	2.78	9.44	41.44	4.56	11.33	74.89
100	AIC'	24.10	62.8	73.40	56.40	71.60	75.70
	BIC'	11.00	30.2	62.00	7.60	33.00	75.50
	ICOMP'	2.90	18.2	64.40	4.50	14.00	72.10

Tablo 5.7 P_{out} for 10% Outlier in Y using GA by Fitness Function . AIC′, BIC′, ICOMP′(IFIM)

		p=4			p=5		
		Kappa Value			Kappa Value		
n	Information C.	3	4	5	3	4	5
20	AIC'	237.50	112.50	75.00	252.00	221.00	150.0
	BIC'	212.00	150.50	47.50	348.00	244.50	168.5
	ICOMP'	1.50	0.50	0.00	2.00	0.50	0.50
30	AIC'	21.00	2.67	0.33	27.67	6.33	0.00
	BIC'	29.67	2.00	0.33	54.67	7.67	5.67
	ICOMP'	0.33	0.33	0.00	1.00	0.00	0.00
40	AIC'	8.00	1.00	0.25	10.00	1.25	0.50
	BIC'	8.00	1.00	0.25	10.00	1.25	0.50
	ICOMP'	1.00	0.75	2.25	1.50	0.00	1.50
50	AIC'	5.80	9.00	19.00	5.20	0.80	0.00
	BIC'	3.00	1.80	8.40	5.00	3.20	2.80
	ICOMP'	0.20	0.40	10.20	2.20	0.80	7.00
60	AIC'	9.17	20.33	28.83	2.50	0.00	0.17
	BIC'	2.00	1.00	3.50	2.83	2.50	9.17
	ICOMP'	2.17	0.67	19.17	1.33	1.67	9.33
70	AIC'	16.00	42.00	33.86	4.14	6.43	27.14
	BIC'	4.00	9.86	29.29	3.00	6.14	20.14
	ICOMP'	3.14	2.71	42.29	2.43	2.00	27.00
80	AIC'	3.25	22.88	44.13	3.63	6.25	34.88
	BIC'	6.13	18.88	43.63	3.50	17.25	35.88
	ICOMP'	4.25	10.00	53.88	2.63	1.13	44.13
90	AIC'	3.67	21.44	61.11	4.22	20.56	38.67
	BIC'	9.22	25.00	51.33	6.22	29.44	48.33
	ICOMP'	3.89	18.22	71.89	4.33	3.78	52.44
100	AIC'	8.00	30.20	59.00	4.30	35.10	59.60
	BIC'	8.80	36.70	59.10	5.50	29.90	52.10
	ICOMP'	4.80	20.50	70.60	4.90	11.90	64.20

These comparisons are graphically presented in from Figure 5.8 to Figure 5.11. Results of experiments are illustrated only the Kappa value for three in Figure 5.8 to Figure 5.11. Since the conclusions are the same for Kappa values four and five, hence the plots of results for these values will not be given here.

Comparisons of performances for detection outliers when sample size is 20 are presented in Figure 5.8 and Figure 5.9. So, it seems that in Figure 5.8(a-h) there is high dependency between dimension of regression model and AIC′and BIC′ information criterion. However, the ICOMP′(IFIM) is not affected by dimension of regression model for detection of outliers. Also, comparisons of performances for detecting of outliers when sample size is n>30 is presented in Figure 510 and 5.11. It seems that in Figure 5.10, there is a relationship between sample size and AIC′ information criterion when ω=10%, in which AIC′ information criterion fails as the sample sizes increases. In comparison the ICOMP′(IFIM) information criteria is far less sensitive to sample sizes when ω=10%.

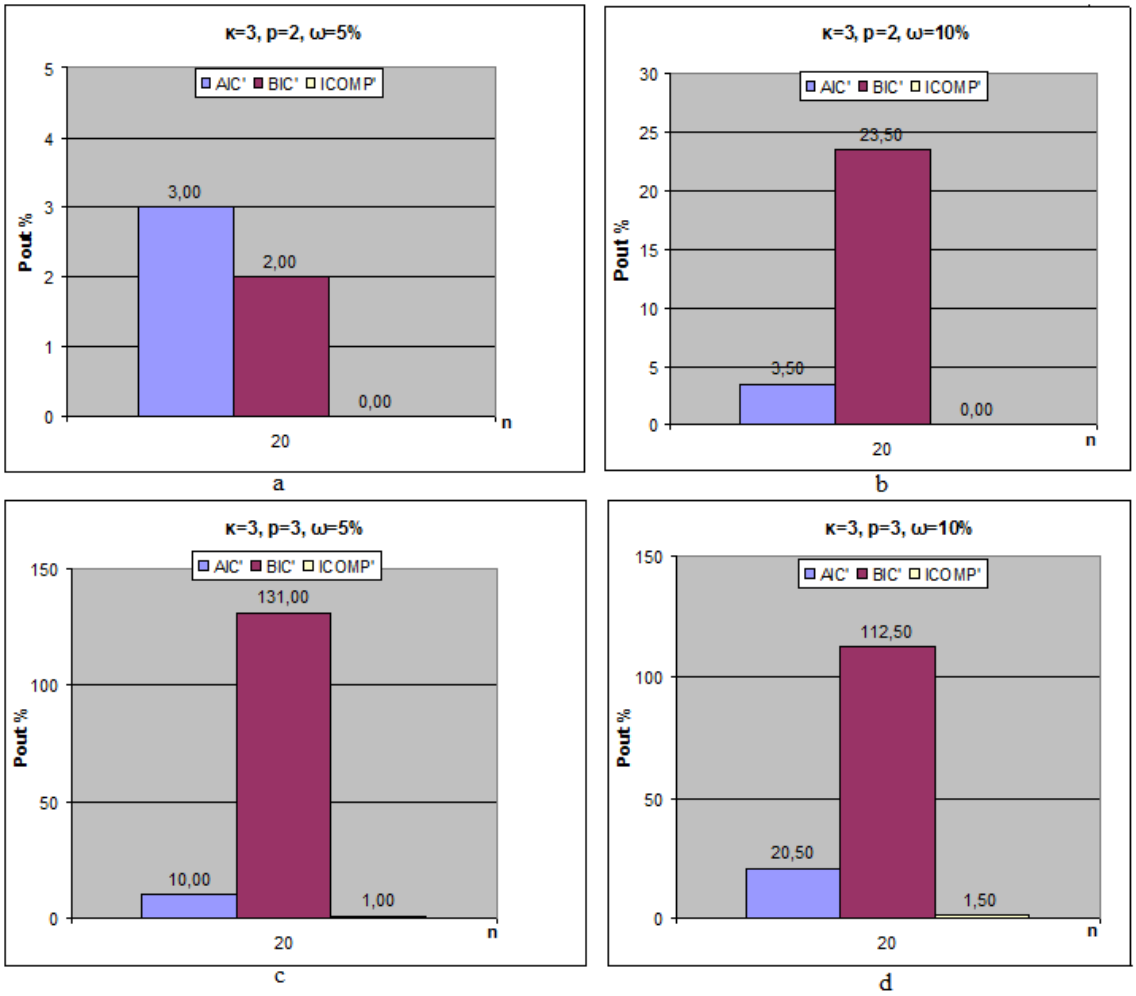

Figure 5.8(a-d) Performance comparisons of information criterion against dimension of p=2,3 when n=20.

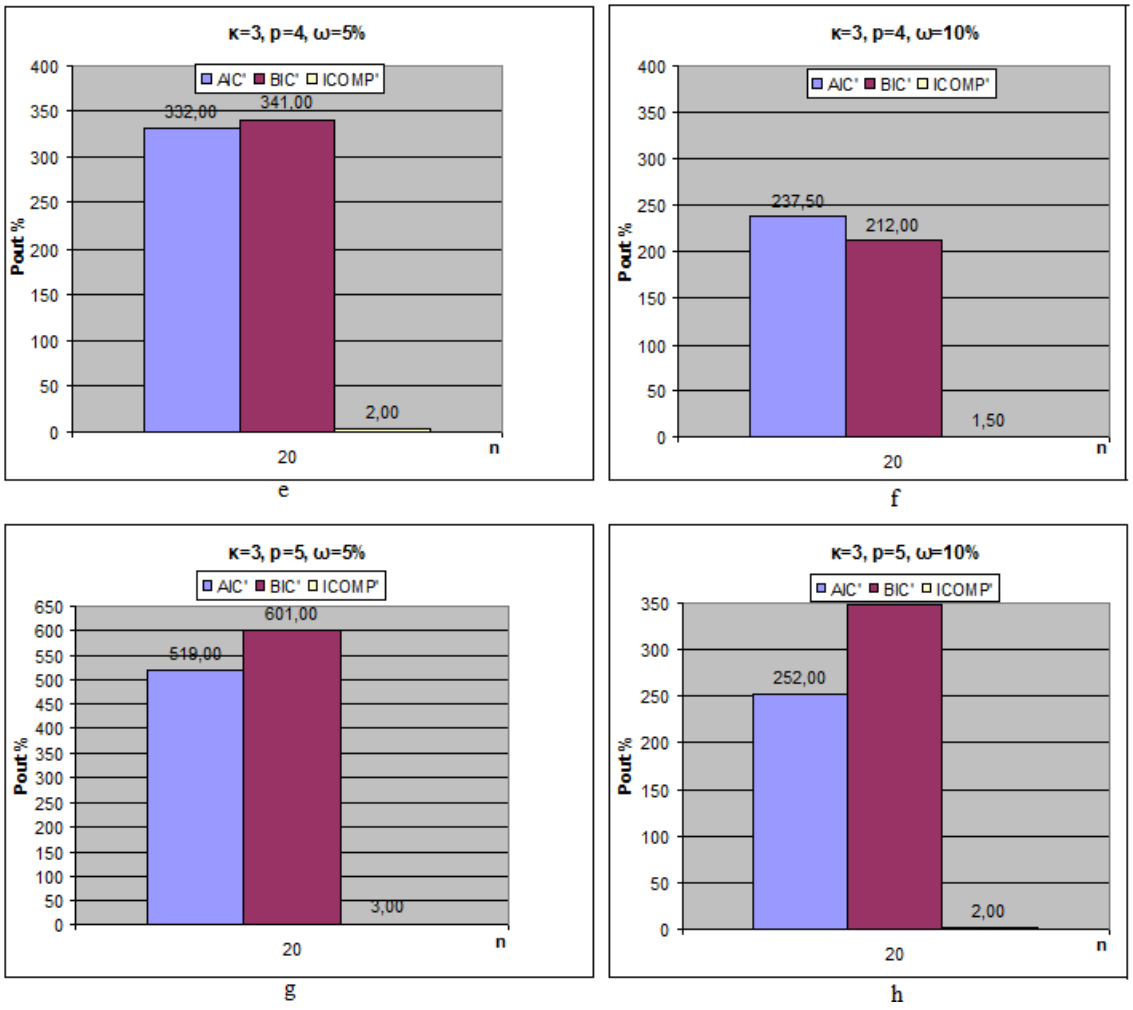

Figure 5.8(e-h). Performance comparisons of information criterion against dimension of p=4,5 when n=20.

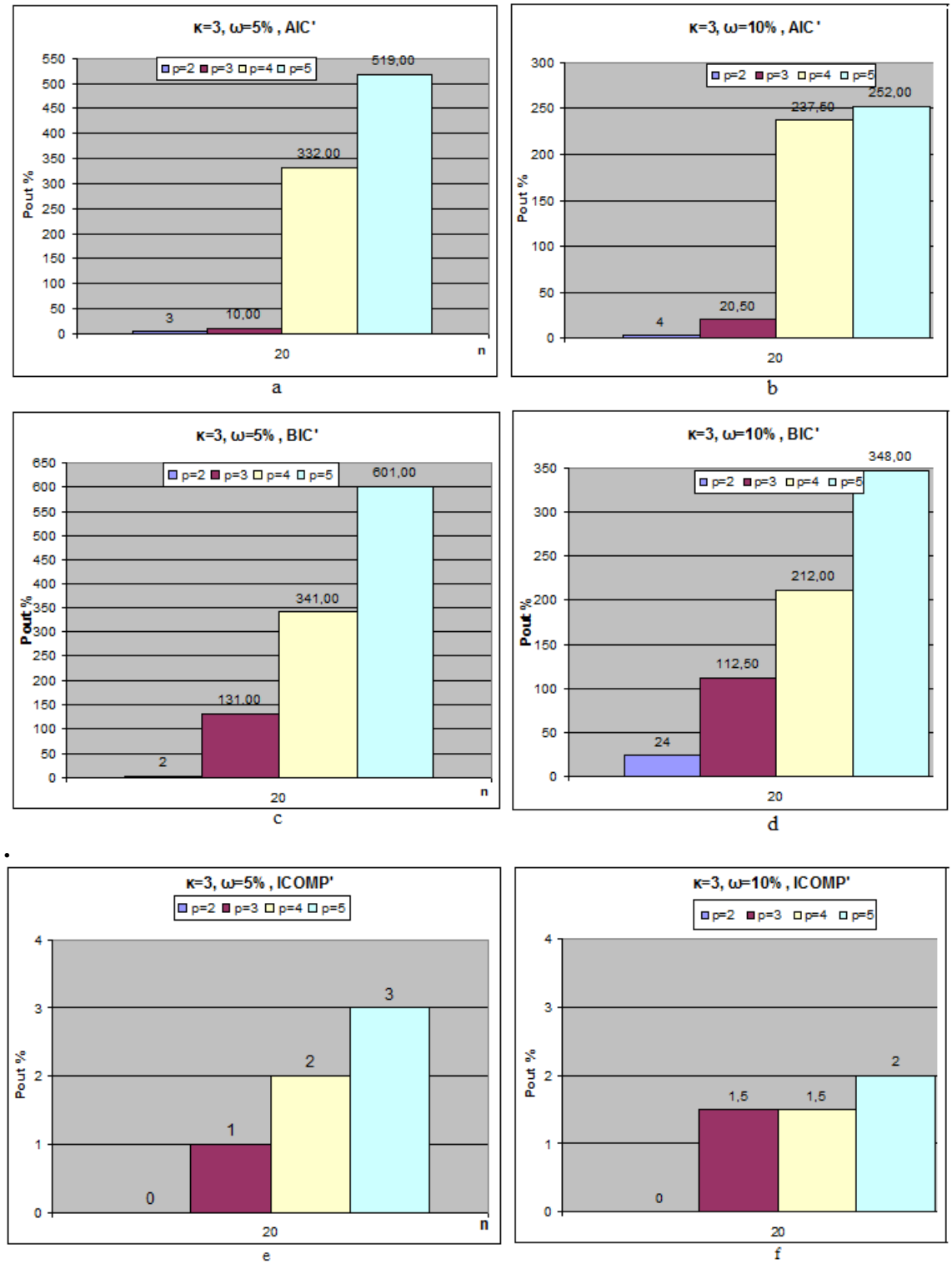

Figure 5.9(a-f). Performance comparison of each information criterion against dimension of regression models when n=20.

It is seen that in Figure 5.9(c-d) the BIC′ criteria is very poor in the detection of outliers for n=20, however the ICOMP' criteria is the robust procedure in detecting the outliers, because the ICOMP' criteria results more consistent than other criteria, they are not

affected by dimension of regression model or percentage of outliers. The results for these figures are summarized as the following:

- Performance of the AIC': The performance of the AIC' for contamination 5% is shown in Figure 5.8 (a-h) and Figure 5.9 (a-b), where it can be noted that this procedure works well under small percentage of outliers and p=2. Also, it performs quite well for contamination 5%, p=2. Moreover, the AIC' has the general failure of the minimum information criterion in all cases for n=20 and it incorrectly selects the observations as outliers.

- Performance of the BIC': The BIC' properly finds outliers for low dimensional data which is p=2 as shown in Figure 5.8(a-d) and Figure 5.9(c-d) However, the BIC' fails in all cases for n=20 than other criteria. It also incorrectly selects the observations as outliers.

- Performance of the ICOMP': The ICOMP' is considered as the consistent method in other criterion, where we clearly see that it works very well across for all dimensional data and all percentages of contaminates ω=5%-10%. Also, it is important to note that the more explanatory variables are selected, the better outlier detection is performed.

The performance of information criterion with different percentages of outliers in Y is depicted in Figure 5.10(a-h) and Figure 5.11(a-f) for $n \geq 30$.

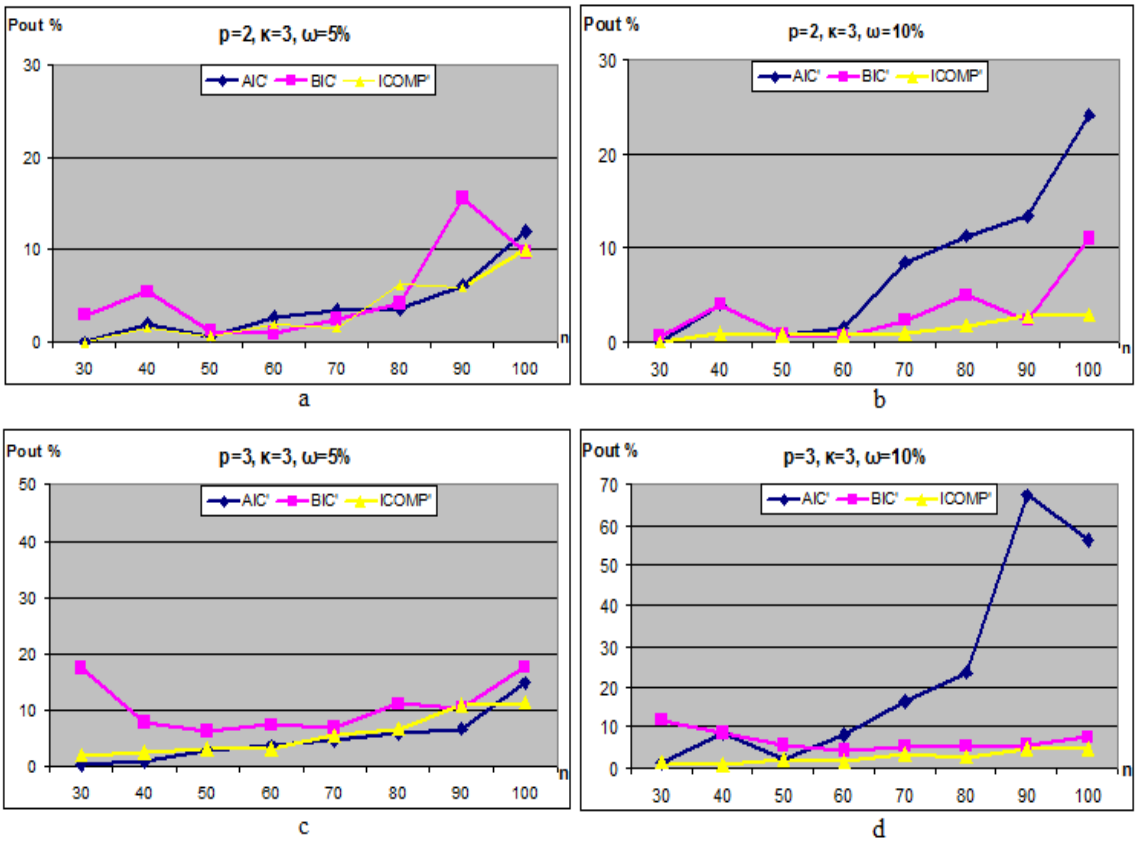

Figure5.10(a-d). Performance comparisons of information criterion against each dimension of regression models p=2,3 when $n \geq 30$.

In Figure 5.10 (a-d), it is plotted the P_{out} values of AIC′, BIC′, and ICOMP′(IFIM) as a function of sample sizes. When n>60, and p=2,3 the P_{out} values of BIC′, and ICOMP′(IFIM) slowly increasing except for AIC' which values are rapidly increases for ω=10%. Moreover BIC′, and ICOMP′(IFIM) curves closely follows for contamination 10% and p=3, while the AIC criteria drastically underestimates outliers.

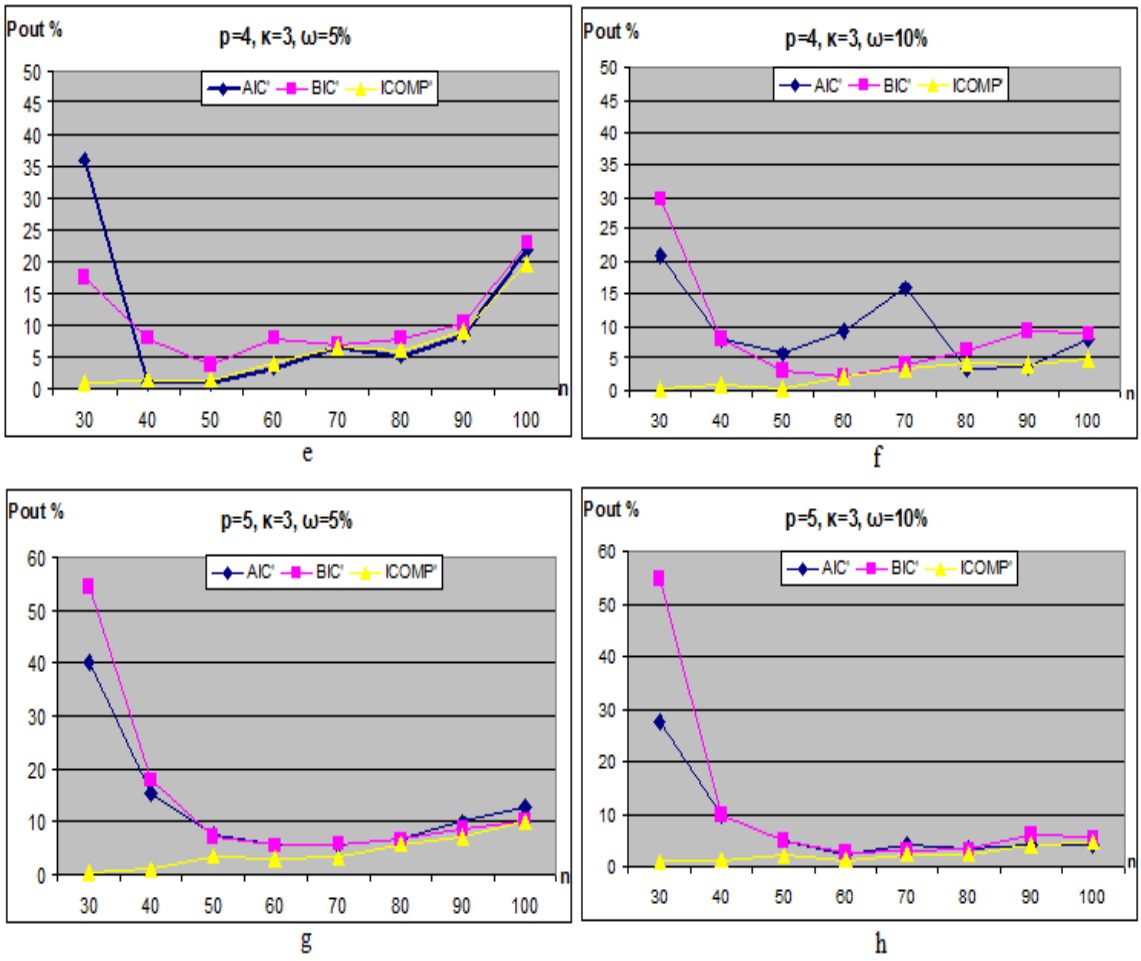

Figure 5.10(e-h). Performance comparisons of information criterion against each dimension of regression models p=4,5 when $n \geq 30$.

It can be clearly observed that from Figure 5.10(e-h) for sample size and dimensions of regression models are increased the AIC′, BIC′, and ICOMP′(IFIM) information criterion more accurate performs to outlier detection. Moreover, the ICOMP' performs better than the other criteria as sample sizes increases. As it is increased the contamination for the same size and dimension of models for p=4, 5, the performance of the AIC', BIC', and ICOMP' information criterion are became more accurate**.**

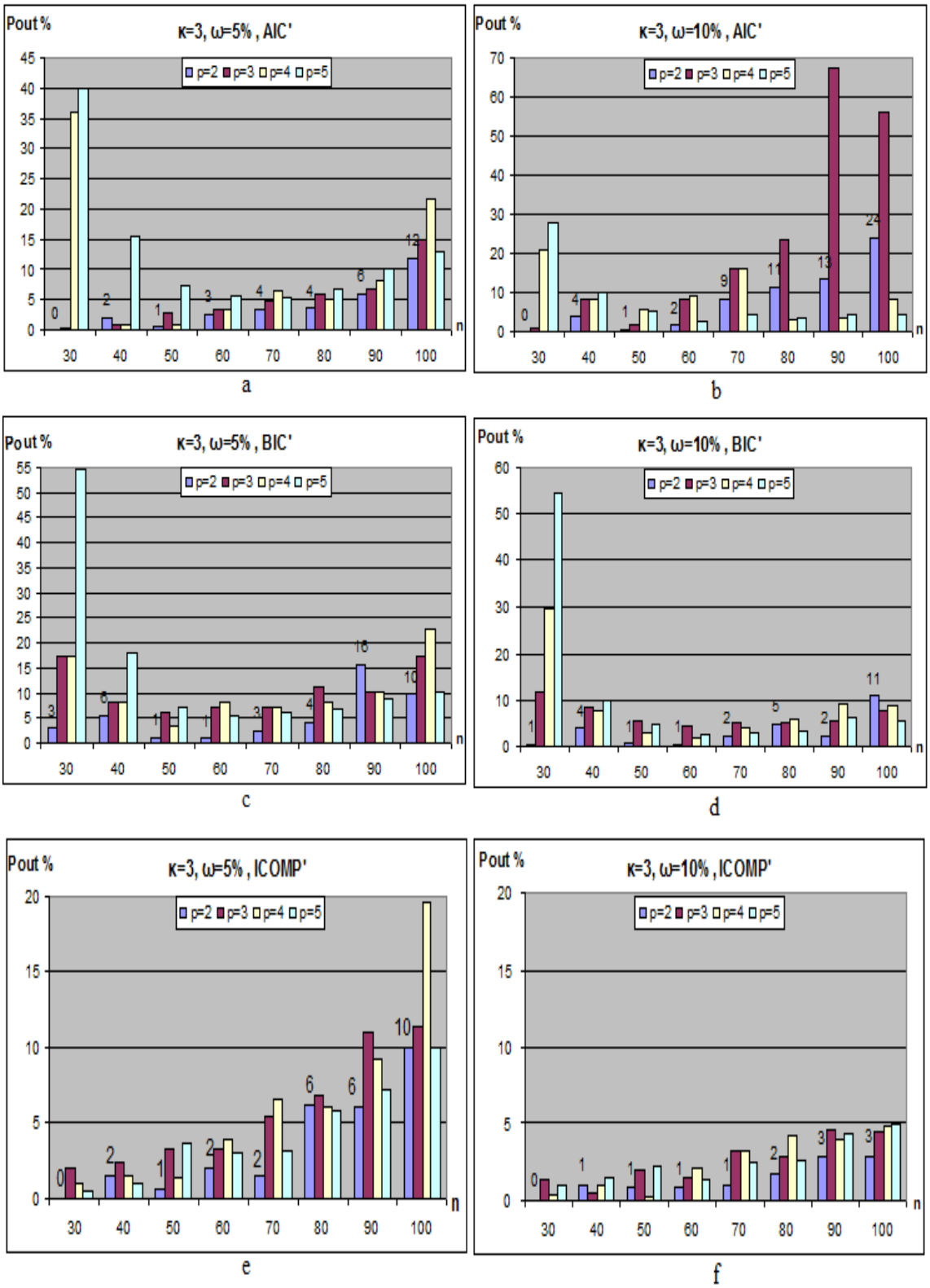

Figure 5.11(a-f). Performance comparison of each information criterion against dimension of regression models when $n \geq 30$.

Performance comparison of information criterion against all dimensions when $n \geq 30$ in Figure 5.11(a-f) are summarized as the following,

- The performance of the AIC' for ω=5-10% is shown in Figure 5.11(a)-5.11(b), where it can be noted that this procedure works well under small percentage of outliers and p=2, 3. Also, it performs quite well for ω=10%, p=4, 5 except for n=30. Moreover, the AIC' is the general selects truly the observations as outliers when small sample sizes and dimension of regression models. The performance of AIC' criteria decrease as the sample size increases.

- The BIC' information criteria is considered truly found outliers for high dimensional data p>2 and ω=5%. However, the performance of BIC' criteria for outlier detection decrease as the sample size and percentage of contamination increase. The BIC' fails for n< 40. It incorrectly selects the observations as outliers for high dimension of regression models.

- The ICOMP' information criteria is considered the most trusted (robust) approach, where we clearly see that is works very well across for all dimensional data and all percentages of contaminates ω=5%-10% than other criteria. However, the sample size and dimension of regression models are increased tend to have larger P_{out} values.

One important result from these figures was the run time of information criterion tends to increase linearly as both the number of observations and the number of outliers are increased.

CHAPTER SIX
CONCLUSIONS

In many data analysis tasks a large number of variables are being recorded or sampled. One of the first steps towards obtaining a coherent analysis is the detection of outlying observations. Although outliers are often considered as an error or noise, they may carry important information. Detected outliers are candidates for aberrant data that may otherwise adversely lead to model misspecification, biased parameter estimation and incorrect results. It is therefore important to identify them prior to modeling and analysis (Williams et al., 2002; Liu et al., 2004). An exact definition of an outlier often depends on hidden assumptions regarding the data structure and the applied detection method. Yet, some definitions are regarded general enough to cope with various types of data and methods. Hawkins (1980) defines an outlier as an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism. Barnett and Lewis (1994) indicate that an outlying observation, or outlier; is one that appears to deviate markedly from other members of the sample in which it occurs.

Identification of the outlying observations allows the disturbance effects are removed. A clean data set is then available for modeling and forecasting purposes. It may happen that, for deeper understanding of the disturbances and for better forecasting, models have to be set up for outliers. The preliminary identification of the effects of the disturbances can help to correctly handle parametric procedures for model building. Outlier detection is a long-lasting issue in regression analysis and the problem will become very complex when it come to multiple regression modeling where outliers, or extreme observations, are on one or a combination of variables. The presence of outliers can exert negative influences on the fit of the multiple regression models. There are some traditional approaches to detecting outliers and they have to be done separately for outlying observations.

The case of unknown location and type of the outlying observations has been considered extensively in the literature for outliers in isolation. The case of multiple

outliers is more difficult to study because of the great number of alternatives and of the masking and swamping effects. If a stretch of n observations is available, the simple alternative if any observation is or not an outlying one generates $2^n - 1$ sets. The tentative multiple regression model is determined using a stochastic search technique termed Genetic Algorithm (GA) with all observations included. Once the tentative model is selected, the predictor subset will stay fixed and will be fitted repeatedly during the process. By using the same subset of predictors, put on an equal scale is the measurement of each observation in the sample. The technique GA is used to select optimal subsets of variables and it represents the presence or absence of a predictor variable using 1 or 0, respectively, on a binary string representing the full multiple regression model. This algorithm makes possible the rapid selection of variable subsets and avoids the computational burden of exhausting all possible combinations of predictor variables.

A genetic algorithm is an optimization tool which mimics the evolution of a population towards fitness to the natural environment. The key feature of a GA is the manipulation of a population whose individuals are characterized by possessing a chromosome. This latter can be coded as a string of characters of given length. Each string represents a feasible solution of the optimization problem. The coding of the individuals depends on the correspondence between strings and chromosomes. A genetic algorithm has been proposed for the identification of the outliers in a multiple regression models. Our rather standard formulation of the algorithm proved to be quite effective in the presence of such a large space of candidate solutions. Notice that in the genetic algorithm iterative procedure, unlike other iterative methods, outliers are not identified and removed one at a time, but, for each chromosome, the fitness is computed on the whole encoded outliers' pattern. This feature seems to be able to cope efficiently with the swamping effect, which arises when observations, that are consistent with most of the data, are yet incorrectly detected as outliers, possibly because of some kind of accidental diversity from that immediately after or before, and with the problem of masking, which is peculiar in this context, where consecutive outliers are very likely to occur in practice.

A simultaneous choice, as in our algorithm, should be more appropriate in this respect. The fitness functions for this GAs are built on using information criteria. They are used both to select the variables of the model, and to determine the outlying observations.

Main contribution of this thesis is to be the first study evaluating the different information criteria for genetic algorithms based outlier detection in multiple regression. We have derived and investigated criterion for outlier detection in multiple regression using genetic algorithms. It seems that the GAs are able to avoid the potential problems of swamping and masking, which sometimes cause problems for reliable outlier detection. In addition, simultaneous outlier detection is possible by GAs.

The core idea behind our approach is to use penalized information criterion to determine an outlier observation for multiple regression. The penalized procedure has the advantage to remove outliers and it does not suffer from masking or swamping problems. Moreover, a large simulation study is undertaken to compare the performance of new approaches (AIC' and ICOMP') to BIC' criteria. Synthetic data are generated for different number of sample sizes and independent explanatory variables. Performances (Scalability) and the efficiency (robustness) of the method for new approaches information criterion are presented. The numerical example and simulation results clearly show a much improved performance of the proposed approach in comparison to existing methods especially followed by applying the ICOMP' approach in order to accurately (robustly) detect the outliers. One important result from these comparisons was that the run time of information criterion tends to increase linearly as both the number of observations and the number of outliers is increased. In the case that the number of observations goes to infinity, the criterion will estimate the right percentage of outliers or even detects properly the most of outliers.

In Figure 5.10 (a-d), it is shown that the plots of the P_{out} values of $AIC', BIC',$ and $ICOMP'(IFIM)$ as a function of sample sizes. When n>60, p=2, 3, the P_{out} values of $BIC',$ and $ICOMP'(IFIM)$ slowly increasing except for AIC' which values are rapidly increases for ω=10%. Moreover $BIC',$ and $ICOMP'(IFIM)$ curves closely follows

for contamination 10% and p=3, while the AIC criteria drastically underestimates outliers.

It can be clearly observed that from Figure 5.10(e-h) for sample size and dimensions of regression models are increased the $AIC', BIC', \text{and } ICOMP'(IFIM)$ information criterion more accurate performs to outlier detection. Moreover, the ICOMP' performs better than the other criteria as sample sizes increases. As it is increased the contamination for the same size and dimension of models for p=4, 5, the performance of the AIC', BIC', and ICOMP' information criterion are became more accurate

The ICOMP' information criteria is considered the most trusted (robust) approach, where we clearly see that is works very well across for all dimensional data and all percentages of contaminates ω=5%-10% than other criteria in Figure 11(a-f). However, the sample size and dimension of regression models are increased tend to have larger P_{out} values.

In conclusion, we note that numerical results clearly demonstrate an excellent performance of ICOMP' criteria as compared to AIC' and BIC' when it is used in outlier detection in multiple regression.

REFERENCES

Acuna, E., & Rodriguez, C. (2004). A Meta analysis study of outlier detection methods in classification, *Technical paper,* Department of Mathematics, University of Puerto Rico at Mayaguez, In *proceedings IPSI 2004*, Venice. http://academic.uprm.edu/~eacuna/paperout.pdf. 02 May 2009

Aggarwal, C. C. & Yu, P. S. (2001). Outlier detection for high dimensional data. *Proceeding. SIGMOD Conference*, 37-46.

Aggarwal, C. C. & Yu, P. S. (2005). An effective and efficient algorithm for high-dimensional outlier detection. *The VLDB Journal*, 14 (2), 211-221.

Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (2005*).* Automatic subspace clustering of high dimensional data*. Data Mining and Knowledge Discovery*, Springer, 11 (1), 5-33.

Agyemang, M., Barker, K., & Alhajj, R. (2006). A comprehensive survey of numeric and symbolic outlier mining techniques. *Intelligent Data Analysis 10 (*6), 521-538.

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. Proceeding. *2nd International Symposium on Information Theory*, 267-281.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transaction on Automatic Control.* 19, 716-723.

Anderson, D., Frivold, T., Tamaru, A., & Valdes, A. (1994). *Next-generation intrusion detection expert system (nides), software users manual, beta-update release.* Technical Report. SRI-CSL-95-07, Computer Science Laboratory, SRI International.

Angiulli, F., Basta, S., & Pizzuti, C. (2006). Distance-based detection and prediction of outliers. *Knowledge and Data Engineering, IEEE Transactions*. 18 (2), 145-160.

Anscombe, F. J. (1960). Rejection of outliers, *Technometrics*, 2, 123-147.

Atkinson, A.C. (1986). Influential observations, high leverage points, and outliers in linear regression. *Statistical Science*, 1, 397-402.

Azzaro-Pantel, C., Bernal-Haro, L., Baudet, P., Domenech S., Pibouleau L. (1998). A two-stage methodology for short term batch plant scheduling: discrete-event simulation and genetic algorithm. *Computers & Chemical Engineering*, 22 (10), 1461-1481.

Bäck, T. (1996). *Evolutionary algorithms in theory and practice*, New York: Oxford University Press.

Baker, J. E. (1985). Adaptive selection methods for genetic algorithms in: *Proceeding. International Conference on Genetic Algorithms and Their Applications*, 101-111.

Barnett, V., & Lewis, T. (1994). *Outliers in statistical data* (3rd ed.). USA: John Wiley & Sons.

Bay, S. D., & Schwabacher, M. (2003). Mining distance-based outliers in near linear time with randomization and a simple pruning rule. *Proc. of the ninth ACM-SIGKDD Int'lConference on Knowledge Discovery and Data Mining*.

Batini, C., & Scannapieca, M. (2006). *Data quality: concepts, methodologies and techniques.* Berlin Heidelberg: Springer-Verlag.

Bayarri, M. J., & Morales, J. (2003). Bayesian Measures of Surprise for Outlier Detection, *Journal of Statistical Planning and Inference*, 111, 3-22.

Bearse, P. M., Bozdogan, H.,. Schlottmann, A. M. (1997). Empirical Econometric Modelling of Food Consumption Using a New Informational Complexity Approach. *Journal of Applied Econometrics*, 12 (5), 563-586.

Beasley, D., Bull, D. R., & Martin, R. R. (1993). An overview of genetic algorithms: Part 1, fundamentals. *University Computing*. 15, 58-69.

Beckman, R. J., & Cook, R. D. (1983). Outliers, *Technometrics*, 25, 119-163.

Bellman, R. (1961). *Adaptive control processes: A guided tour.* New-Jersey: Princeton University Press.

Belsley D. A., Kuh E., & Welsch R. E. (1980). *Regression diagnostics: identifying influential data and source of collinearity*. New York: John Wiley.

Bishop, C. M. (1994). Novelty detection and neural network validation. *Proceedings of the IEE Conference on Vision, Image and Signal Processing*. 141, 217–222.

Booker, L. B., Fogel, D. B., Whitley, D. & Ageline, P. J. (1997). *Recombination in: The Hanbook of evolutionary computation,* T. Bäck, D.B. Fogel, and Z. Michalewicz, chapter E3.3, pp.C3.3:1-C3.3:27. Philadelphia: IOP Publishing and Oxford University Press.

Breuning M. M., Kriegel H.P., Ng R.T., & Sander J. (2000). LOF: Identifying density-based local outliers. *Proceeding SIGMOD Conference,* 93-104.

Boriah, S., Chandola, V., & Kumar, V. (2008). Similarity measures for categorical data: A comparative evaluation. In *Proceedings of the eighth SIAM International Conference on Data Mining*. 243-254.

Bozdogan, H. (1987). Model-selection and Akaike's information criterion (AIC): The general theory and its analytical extensions. *Psychometrika*, 52, 345-370.

Bozdogan, H., & Haughton D. M. A. (1998). Informational Complexity Criteria for Regression Models*, Computational Statistics and Data Analysis*, 28, 51-76.

Bozdogan, H. (2000). Akaike's Information Criterion and Recent Developments in Information Complexity. *Journal of Mathematical Psychology*, 44, 62-91.

Bozdogan H. & Bearse P. (2003). Information complexity criteria for detecting influential observations in dynamic multivariate linear models using the genetic algorithm. *Journal of Statistical Planning and Inference*, 114, 31-44.

Bozdogan, H. (2004). *Statistical Data Mining and Knowledge Discovery*. USA : Chapman and Hall/CRC.

Caussinus, H., & Roiz, A. (1990). Interesting projections of multidimensional data by means of generalized component analysis. *Physica*, 90, 121-126.

Chandola V., Banerjee, A., & Kumar V. (2007). *Anomaly Detection: A Survey*. Technical Report. Department of Computer Science and Engineering University of Minnesota, TR 07-017.

Chandola, V., Boriah, S., & Kumar, V. (2008). Understanding categorical similarity measures for outlier detection. Technical Report. 08-008, University of Minnesota.

Chen G., Chen, S., Guo, W., & Chen H. (2007). The multi-criteria minimum spanning tree problem based genetic algorithm, *Information Sciences*, 177 (22). 5050–5063.

Cook, R. D. (1979). Influential Observations in Linear Regression. *Journal of the American Statistical Association*, 74, 169-174.

Crawford, K. D., & Wainwright, R. L. (1995). Applying Genetic Algorithms to outlier detection, *Proceedings of the Sixth International Conference on Genetic Algorithms*, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.33.7258.

Daniel, C. (1960). Locating outliers in factorial experiments. *Technometrics*, 2, 149-156.

Davis, L. (1991). *Handbook of genetic algorithms*. New York: Van Nostrand Reinhold.

Dawkins, R. (1996). *The blind watcmaker: why the evidence of evolution reveals a universe without design*. USA:W. W. Norton.

Desforges, M., Jacob, P., & Cooper, J. (1998). Applications of probability density estimation to the detection of abnormal conditions in engineering. *Proceedings of Institute of Mechanical Engineers*. 212, 687-703.

DuMouchel, W., & Schonlau, M. (1998). A fast computer intrusion detection algorithm based on hypothesis testing of command transition probabilities. *Proceedings of the 4th International Conference on Knowledge Discovery and Data-mining* (KDD98), 189-193.

Eiben, A. E., & Smith, J. E. (2003). *Introduction to Evolutionary Computing*, Bristol: Springer.

Endler, D. 1998. Intrusion detection: Applying machine learning to solaris audit data. In *Proceedings of the 14th Annual Computer Security Applications Conference*. IEEE Computer Society, 268-279.

Eshelman, L. J., Caruana, R. A., & Schaler J. D. (1989). Biases in the Crossover Landscape, in: *Proceeding 3rd International conference on genetic algorithms*, 10-19, Morgan Kaufmann Publishers, San Mateo.

Eskin, E. (2000). Anomaly detection over noisy data using learned probability distributions. *Proceedings of the Seventeenth International Conference on Machine Learning*. Morgan Kauf-mann Publishers. 255-262.

Eskin, E., Arnold, A., Prerau, M., Portnoy, L., & Stolfo, S. (2002). A geometric framework for unsupervised anomaly detection. In *Proceedings of Applications of Data Mining in Computer Security*. Kluwer Academics, 78-100.

Ester, M., Kriegel, H.P., Sander, J., Xu, X. (1998). Clustering for mining in large spatial databases, *KI-Journal (Artificial Intelligence), Special Issue on Data Mining*, 12 (1), 18-24.

Fawcett, T., & Provost, F. (1997), Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3), 291–316.

Fawcett, T., & Provost, F. J. (1999). Activity monitoring: noticing interesting changes in behavior. *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 53–62.

Fogel, D. B. (1998). *Evolutionary Computation: The Fossil Record*. Piscataway: IEEE Press.

Forrest, S. (1993). Genetic algorithms: principles of natural selection applied to computation. Science, 261, 872-878.

Fox, J. (1997). *Applied regression analysis, linear models and related methods* (3rd ed). Sage Publication, USA.

Gen, M., Cheng R. (2000). *Genetic algorithms and engineering optimization*. USA: Wiley

Gnanadesikan, R., & Kettering, J. (1972). Robust estimates, residuals and outlier detection with multi-response data. *Biometrics*, 28, 81-124.

Goldberg D. E. (1989). *Genetic algorithm in search, opimization, and machine learning*. New York: Addison-Welsey.

Goldberg, D. E., & Deb K. (1991). *A Comparative analysis of selection schemes used in genetic algorithms*. In Foundations of Genetic Algorithms, San Mateo, California, USA: Morgan Kaufmann Publishers, 69-93.

Golberg, D. E. (2002). Design of Innovation. *Lessons from and for competent genetic algorithms*. Boston: Kluwer.

Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101 (23), e215-e220. http://circ.ahajournals.org/cgi/content/full/101/23/e215.

Goonatilake, S., & Treleaven, P. (1995). *Intelligent systems for finance and business.* New York: John Wiley & Sons.

Graham, R., D. (2002). Radio emerges from the electronic soup. *New Scientist*, 175 (2358), 19.

Grefenstette, J. J., & Baker, J. E. (1989). How genetic algorithms work: a critical look at implicit parallelism, in: *Proceeding 3rd international Conference on Genetic Algorithms*, 20-27.

Grubbs, F. E. (1969). Procedures for Detecting Outlying Observations in Samples. *Technometrics*, 11, 1–21.

Grupe, F. H., & Jooste, S. (2004). Genetic algorithms: a business perspective. *Information Management and Computer Security*, 12, 289-298.

Guttman, I. (1973). Care and handling of univariate or multivariate outliers in detecting spuriosity a Bayesian approach. *Technometrics*, 15, 723-738.

Hadi, A. (1986). Influential observations, high leverage points, and outliers in linear regression. *Journal of the American Statistical Association, Statistical Science*, 1 (3), 379-393.

Hadi, A. S. (1992). Identifying multiple outliers in multivariate data. *Journal of the Royal Statistical Society*. Series B, 54, 761-771.

Hair, J., Anderson, R., Tatham, R. & Black, W. (1998). *Multivariate data analysis.* New- Jersey: Prentice Hall International.

Haupt, R., & Haupt, S., E. (1998). *Practical genetic algorithms*. New-York: John Wiley & Sons.

Hawkins, D. (1980). *Identification of outliers*. London: Chapman and Hall.

Hawkins, S., He H. X., Williams G. J., & Baxter R. A. (2002). Outlier detection using replicator neural networks. *Proceedings of the Fifth International Conference and Data Warehousing and Knowledge Discovery* (DaWaK02).

He, Z., Xu, X., & Deng, S. (2003). Discovering cluster-based local outliers. *Pattern Recognition Letters 24,* 9 (10), 1641-1650.

Hoaglin, D. & Tukey, J. (1983). *Understanding robust and exploratory data analysis*. Canada: JohnWiley.

Hoeting, J., Raftery, A., & Madigan, E. D. (1996). A method for simultaneous variable selection and outlier identification in linear regression. *Computational Statistics and Data Analysis,* 22, 251-270.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Michigan:The MIT Press.

Hu, T., & Sung, S. Y., (2003). Detecting pattern-based outliers. *Pattern Recognition Letters*, 24, 3059-3068.

Hurvich, C. F., & Tsai C.L. (1991). Regression and Time Series Model Selection in Small Samples, *Biometrika*, 76, 499-509.

Iglewicz, B., Hoaglin D. C. (1993) How to detect and handle outliers. *ASQ basic References in Quality Control*, 16.

Ishibuchi H., Nakashima T., Nii M. (2001). *Genetic Algorithm based instance and feature selection*. *Instance Selection and Construction for Data Mining*, Huan Liu, Hiroshi Motoda, *Springer*, 95-112.

Jann, A. (2000). Multiple change point detection with a Genetic Algorithm. *Soft Computing*, 4 (2), 68-75.

Japkowicz, N., Myers, C., & Gluck M. A. (1995). A Novelty Detection Approach to Classification. *Proceedings of the 14th International Conference on Artificial Intelligence* (IJCAI-95), 518–523.

Jin, W., Tung A., & Han, J. (2001). Mining top-*n* local outliers in large databases, *Proceedings of the 7th International Conference on Knowledge Discovery and Datamining* (KDD01).

Judd, K. (2003) Building optimal models of time series, in Chaos and Its Reconstruction, eds. Gouesbet, G., Meunier-Guttin-Cluzel, S. & Menard, O. *Nova Science Publication*, 179–214.

Kaya, A. (2004). Outlier effects on databases. *Advances in Information Systems, Springer*, 3265(2005), 88-95.
http://www.springerlink.com/content/er1xt04wbx180903. 30 April 2009.

Kitagawa, G. & Akaike, H. (1982). A quasi Bayesian approach to outlier detection. *Annals of the Institute of Statistical Mathematics*. 34 (1), 389-398.

Kitagawa, G. (1984). Bayesian analysis of outliers via Akaike's predictive likelihood of a model. *Communications in Statistics - Simulation and Computation*, 13 (1), 107-126.

Knorr, E., & Ng, R. (1997). A unified approach for mining outliers. In *Proceedings Knowledge Discovery* KDD, 219–222.

Knorr E. & Ng R. (1998). Algorithms for mining distance-based outliers in large datasets. Proceding of VLDB'98, 392-403.

Knorr, E., M. (2002). *Outliers and data mining: finding exceptions in data*. Doctor of Philosophy Thesis, The University of British Columbia.

Kohonen, T. (1997). *Self-Organizing Maps*. Berlin: Springer-Verlag.

Kollios G., Gunopulos D., Koudas N., Berchtold S. (2003), Efficient biased sampling for approximate clustering and outlier detection in large data sets. *IEEE Transactions on Knowledge and Data Engineering*, 15 (5), 1170-1187.

Konishi, S., & Kitagawa G. (2008). *Information Criteria and Statistical Modelling*. New York:Springer.

Kou, Y., Lu, C. T., & Chen, D. (2006). Spatial weighted outlier detection. In *Proceedings of SIAM Conference on Data Mining*.

Kroenke, D. (2003). *Database concept.* New Jersey: Prentice Hall

Kullback, S. & Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22, 79-86.

Kullback, S. (1997). *Information Theory and Statistics*. USA: Dover Publications.

Kwon, S. H., Ueno, M., & Sugeno, M. (1998). A Consistent and bias corrected extension of Akaike's information criterion (AIC): $AIC_{bc}(k)$, *J.KSIAM*, 2 (1), 41-60.

Last, M., & Kandel, A. (2001). Automated detection of outliers in real-world data. *Proceedings of the Second International Conference on Intelligent Technologies*. http://www.ise.bgu.ac.il/faculty/mlast/papers/outliers2.pdf 30 April 2009.

Laurikkala, J., Juhola, M. & Kentala, E. (2000). Informal identification of outliers in medical data. *Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology IDAMAP-2000*.

Lee, K. Y., & El-Sharkawi, M. A. (2008). *Modern heuristic optimization techniques theory and applications to power systems*. Canada: John Wiley & Sons.

Liu H., Shah S., & Jiang W. 2004. On-line outlier detection and data cleaning. *Computers and Chemical Engineering*, 28, 1635-1647.

Lu, C., Chen, D., Kou Y. (2003) Algorithms for spatial outlier detection. In *Proceedings of the 3rd IEEE International Conference on Data-mining* (ICDM'03), Melbourne.

Maaranen, H., Miettinen, K., & Penttinen, A. (2007). On initial populations of a genetic algorithm for continuous optimization problems. *Journal of Global Optimization,* 405–436.

Mangano, S. (May, 1995). *Genetic algorithms: A tutorial*. Retrieved May 20, 2009, from http://deron.csie.ncue.edu.tw/oop/GATutorial_deron.pdf

Markou, M., & Singh, S. (2003). Novelty detection: A review-part I: Statistical approaches. *Signal Processing*, 83 *(*12), 2481-2497.

Maronna, R. D., Martin, R. D. & Yohai, V. J. (2006). *Robust statistics: theory and methods*. London: John Wiley.

Marsland, S. (2001). *On-Line Novelty Detection Through Self-Organization, with Application to Inspection Robotics*. Ph.D. Thesis, Faculty of Science and Engineering, University of Manchester, UK.

McQuarrie, A., Shumway, R., & Tsai, C. L., (1997). The Model Selection Criterion AIC_u, *Statistics and Probability Letters*, 34, 285-292.

Mendenhall, W., Reinmuth, J. E., & Beaver, R. J. (1993). *Statistics for management and economics*. Belmont: CA: Duxbury Press.

Michalewicz, Z. (1996). *Genetic algorithms + data structure = evolution programs*, Berlin: Springer.

Mitchell, M. (1996). *An introduction to genetic algorithms.* MIT Press, Cambridge.

Mitchell, M. (1999). *An Introduction to genetic algorithms* (5^{th} ed).Massachusetts: The Mit Press.

Osborne, J. W., & Overbay, A. (2004). The power of outliers. *Practical Assessment, Research & Evaluation*, 9(6).

Otey, M., Parthasarathy, S., Ghoting, A., Li, G., Narravula, S., & Panda, D. (2003). Towards nic-based intrusion detection. In *Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, New York, NY, USA, 723-728.

Papadimitriou, S., Kitawaga, H., Gibbons, P.G., & Faloutsos, C. (2002). LOCI: Fast Outlier Detection Using the Local Correlation Integral. *Intel Research Laboratory. Technical report no.* IRP-TR-02-09

Patcha, A., & Park, J. M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Elsevier Computer Networks*. 51 (12), 3448-3470.

Pires, A. & Santos-Pereira, C. (2005). Using clustering and robust estimators to detect outliers in multivariate data. In: *Proceedings of International Conference on Robust Statistics*. Finland.

Pyle, D. (1999). *Data Preparation for data mining,* USA: Morgan Kaufmann Publishers.

Ramakrishnan, R., & Gehrke, J. (2000) *Database management systems*. (2nd ed.) USA: McGraw-Hill Higher Education.

Ramaswamy, S., Rastongi, R., & Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. *Proceeding of the ACM SIGMOD Conference*, 427-438.

Reeves, C. R., & Rowe, J. E. (2003). Genetic *algorithms principles and perspectives, a guide to GA theory*. London: Kluwer Academic Publishers.

Roberts, S., & Tarassenko, L. (1994). A Probabilistic resource allocating network for novelty detection. *Neural Computation*, 6 (2), 270–284.

Rothlauf, F. (2006). *Representations for genetic and evolutionary algorithms.* Netherlands: Springer.

Rousseeuw P. J., & Leroy A.M. (1987). *Robust regression and outlier detection*. New York: Wiley-Interscience.

Rousseeuw, P. & Leroy, A. (1996). *Robust regression and outlier detection (*3rd ed.). New York: John Wiley & Sons.

Ruts, I. & Rousseuw, P. (1996). Computing depth contours of bivariate points cloud. *Computational Statistics and Data Analysis*, 23,153-168.

Salvador, S., & Chan, P. (2003). *Learning states and rules for time-series anomaly detection*. Technical Report. CS-2003-05, Department of Computer Science, Florida Institute of Technology Melbourne FL 32901.

Sastry, K., Goldberg, D., & Kendall, G. (2005). Genetic Algorithms. In :Burke E., Kendall G., Search Methodologies: *Introductory Tutorials in Optimization and Decision Support Techniques*. USA: Springer.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6 (2), 461-464.

Shekhar, S., Lu, C. T., & Zhang, P. (2001). Detecting graph-based spatial outliers: algorithms and applications (a summary of results). In *Proceedings of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, New York, NY, USA, 371-376.

Shekhar, S. & Chawla, S. (2002). *A Tour of spatial databases*, Prentice Hall.

Shono, H. (2000). Efficiency of the finite correction of Akaike's Information Criteria, *Fish Sci*, 66 (3), 608-610.

Sinha S. Kumar. (1997). *Sequential Application Of Multivariate Outlier Test: A Robust Approach*. Master Thesis, At Dalhousie Uxiversity Halifax. Nova Scotia.

Snyder, D. (2001). *Online intrusion detection using sequences of system calls*. M.S. thesis, Department of Computer Science, Florida State University

Song L., & Donald E. B. (2002). *Outlier-based data association: combining olap and data mining*. Technical Report, Department of Systems Engineering University of Virginia, SIE 020011.

Song, X., Wu, M., Jermaine, C., & Ranka, S. (2007). Conditional anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 19 (5), 631-645.

Spears, W. M. & De Jong K. A. (1994). On the virtues of parameterized uniform crossover, in: *Proceeding 4th International Conference on Genetic Algorithms*.

Spears, W. (1997). Recombination parameters in: *The Hanbook of evolutionary computation*, T. Bäck, D.B. Fogel, and Z. Michalewicz, chapter E1.3, E1.3:1-E1.3:13. Philadelphia: IOP Publishing and Oxford University Press.

Suárez Rancel M. M., & González Sierra M. A. (1999). Measures and procedures for the identification of locally influential observations in linear regression. *Communications in Statistics and Theory Methods*, 28 (2), 343-366.

Sugiura, N. (1978). Further analysis of the data by Akaike's information criterion and the Finite Corrections. *Communications in Statistics and Theory Methods*, A7, 13-26.

Syswerda, G. (1989). Uniform crossover in genetic algorithms, in: *Proceeding 3rd International Conference on Genetic Algorithms*, 2-9.

Tan, P. N., Steinbach, M., & Kumar, V. (2005). *Introduction to Data Mining*. Addison-Wesley.

Tax, D. M. J. (2001). *One-class classification; concept-learning in the absence of counter-examples*. Ph.D. thesis, Delft University of Technology.

Thomson, W. R. (1935). On a criterion for the rejection of observations and the distribution and the sample mean in samples of n from a normal universe. *Biometrika*, 32, 301-310.

Ting J. A., Souza D., & Schaal S. (2007). Automatic Outlier Detection: A Bayesian Approachi *IEEE International Conference on Robotics and Automation*, Roma-Italy, 2489-2494.

Tolvi J. (2004*).* Genetic algorithms for outlier detection and variable selection in linear regression models, *Soft Computing*, *Springer*, 8 (8), 527-533.

Toroslu, I. H., Arslanoglu, Y. (2007). Genetic algorithm for the personnel assignment problem with multiple objectives. *Information Sciences,* 177 (3). 787-803.

Torr, P., & Murray, D. (1993). Outlier detection and motion segmentation. *In Proceedings of SPIE, Sensor Fusion VI, Paul S. Schenker*, 2059, 432-443.

Uğur, A. (2008). Path planning on a cuboid using Genetic Algorithms. *Information Sciences*, 178(16), 3275-3287.

Van Emden, M. H. (1971). An Analysis of Complexity. *Mathematical Centre Tracts*, 35, Amsterdam.

Varbanov, A. (1998). Bayesian approach to outlier detection in multivariate normal samples and linear models. *Communications in Statistics- Theory and Methods*, 27 (3), 547-557.

Victoria J. H., & Austin J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22 (2), 85–126.

Weigend, A. S., Mangeas, M., & Srivastava, A. N. (1995). Nonlinear gated experts for time-series discovering regimes and avoiding over fitting. *International Journal of Neural Systems,* 6 (4), 373-399.

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing.* 4, 65–85.

Williams, G. J., Baxter, R. A., He H., X., Hawkins S. and Gu L. (2002). A comparative study of RNN for outlier detection in data mining. I*EEE International Conference on Data-mining (ICDM'02)*, Maebashi City, Japan, *CSIRO Technical Report CMIS*-02/102.

Williams, G. J., & Huang, Z. (1997). Mining the knowledge mine: The hot spots methodology for mining large real world databases. *Lecture Notes in Artificial Intelligence,* 1342, 340–348.

Wright, A., H. (1991). Genetic algorithms for real parameter optimization. *Foundations of Genetic Algorithms*. San Mateo: Morgan Kaufman Publishers, 205-218.

Wu, M., & Jermaine, C. (2006). Outlier detection by sampling with accuracy guarantees. *In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, New York, NY, USA, 767-772.

Yamanishi, K., Takeuchi, J. I., Williams, G., & Milne, P. (2004). On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery*, 8, 275-300.

Yang, Y., 2003. Can the strengths of AIC and BIC be Shared?, Department of Statistics Iowa State University, http://www.stat.iastate.edu/preprint/articles/2003-10.pdf. 3 May 2009.

Appendices - 1 Matlab Codes for Outlier Detection in Multiple Regression using Information Criteria for p=5.

1.1 GA Procedure

```
clc
clear all
close all

global varX
global varY
global DegiskenSayisi
global savedScores;
global outlier;
global deg2;

M1=xlsread('x1');
M2=xlsread('x2');
M3=xlsread('x3');
M4=xlsread('x4');
M5=xlsread('x5');
M6=xlsread('y');

for i=1:100

   varX=[M1(:,i) M2(:,i) M3(:,i) M4(:,i) M5(:,i)];
   varY=M6(:,i);.

options = [];

savedScores = [];


 Plot options
options = gaoptimset(options, 'PlotFcns', {@gaplotbestindiv, ... % the best individual
                                @gaplotbestf, ... % the best function value
                                @gaplotscores, ... % the scores
                                @gaplotstopping}); % the stopping criteria

% Population options

options = gaoptimset(options, 'PopulationType', 'bitString');

options = gaoptimset(options, 'PopulationSize', 40, …
             'EliteCount', 2);

% Display options

options = gaoptimset(options, 'Display', 'iter');
options = gaoptimset(options, 'CrossoverFcn',@crossoversinglepoint);
```

```
% Modifying the stopping criteria

options = gaoptimset(options, 'Generations', 250, ...
                'StallGenLimit', 100, ...
                'StallTimeLimit', 60000);

DegiskenSayisi =length(varY);

[katsayilar,fval,reason,output] = ga(@tolvifitness,DegiskenSayisi,options);

fprintf(1,'\n');

for k=1:DegiskenSayisi

   degiskenler(k) = bin2dec(num2str(katsayilar(k)));

end

if sum(degiskenler) > (length(degiskenler) /2)

   degiskenler = 1 - degiskenler;
end

for k=1:DegiskenSayisi

   if degiskenler(k) > 0
      fprintf(1,'bulunan: %u. ornek bir outlier .\n',k);
      xlswrite('k',k);
       deg2 = [deg2 k];
   end
end

outlier=[];
outlier=[outlier deg2'];

savedScores = reshape(savedScores, [40 length(savedScores(:))/40]);
savedScores = savedScores';
figure
subplot(211), plot(savedScores,'.')
subplot(212), plot(min(savedScores,[],2),'.')
xlswrite('saveScores',savedScores);
end
```

1.2 Fitness Function of GA

```
function scores=tolvifitness(katsayilar)

global varX
global varY
global DegiskenSayisi
```

```
global savedScores;

n=DegiskenSayisi;

degiskenler = [];
deg2 = [];

for k=1: n

   if katsayilar(k)
      sutun=zeros(n,1);
      sutun(k,1)=1;
      degiskenler = [degiskenler sutun];

   end

end

 [temp p]=size(varX);

size(degiskenler)

if sum(size(degiskenler)) > 0
   X = [ones(size(varY)) varX degiskenler];
else
   X = [ones(size(varY)) varX];
end

A=pinv(X)*varY;

A=A(:);

temp=varY-X*A;

%vartah=temp'*temp/n;

%scores=log(temp'*temp/n)+(p+1)*log(n)/n+3*sum(katsayilar)*log(n)/n;

%savedScores = [savedScores scores];

%CFisher=(p+1)*log((trace(vartah*(X'*X)^-1)+2*vartah^2/n)/p+1)-
                log(det(vartah*(X'*X)^-1))-log(2*vartah^2/n);

%scores=n*log(2*pi)+n*log(vartah)+n+CFisher+3*sum(katsayilar)*log(n);

 scores= n*log(2*pi)+n*log(temp'*temp/n)+ n+2*(p+1)+3*sum(katsayilar)*log(n);

savedScores = [savedScores scores];
```