APPLICATION OF DATA MINING TECHNIQUES IN BIOINFORMATICS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology
In
Computer Science Engineering

By

AMARTYA SINGH
&
KUNDAN KUMAR DAS

Department of Computer Science and Engineering
National Institute of Technology
Rourkela
2007

APPLICATION OF DATA MINING TECHNIQUES IN BIOINFORMATICS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology
In
Computer Science and Engineering

By

AMARTYA SINGH
&
KUNDAN KUMAR DAS

Under the guidance of
Prof. S.K. Rath

Department of Computer Science and Engineering
National Institute of Technology
Rourkela
2007

National Institute of Technology

Rourkela

CERTIFICATE

This is to certify that the thesis entitled, "**APPLICATION OF DATA MINING TECHNIQUES IN BIOINFORMATICS**" submitted by Sri Amartya Singh and Sri Kundan Kumar Das in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Computer Science Engineering at the National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other university / institute for the award of any Degree or Diploma.

Date: May 15, 2007

Prof S.K.Rath
Dept. of Computer Science Engineering
National Institute of Technology, Rourkela
Rourkela - 769008

ACKNOWLEDGEMENT

I express my sincere gratitude and indebtedness to Dr S.K.Rath , Professor in the department of Computer Science Engineering, NIT Rourkela for giving me the opportunity to work under him and extending every support possible at each stage of this project work. The level of flexibility offered by him in implementing the project work is highly applaudable.

I would also like to convey my deep regards to all other faculty members and staff of Department of Computer Science Engineering, NIT Rourkela, who have bestowed their great effort and guidance at appropriate times without which it would have been very difficult on my part to finish this project work.

Date: May 15, 2007

Amartya Singh
Roll No:10306012
&

Kundan Kumar Das
Roll No: 10306025

CONTENTS

ABSTRACT

With the widespread use of databases and the explosive growth in their sizes, there is a need to effectively utilize these massive volumes of data. This is where data mining comes in handy, as it scours the databases for extracting hidden patterns, finding hidden information, decision making and hypothesis testing. Bioinformatics, an upcoming field in today's world, which involves use of large databases can be effectively searched through data mining techniques to derive useful rules.

Based on the type of knowledge that is mined, data mining techniques [1] can be mainly classified into association rules, decision trees and clustering. Until recently, biology lacked the tools to analyze massive repositories of information such as the human genome database [3]. The data mining techniques are effectively used to extract meaningful relationships from these data. Data mining is especially used in microarray analysis which is used to study the activity of different cells under different conditions.

Two algorithms under each mining techniques were implemented for a large database and compared with each other.

1. Association Rule Mining: - (a) a priori (b) partition
2. Clustering: - (a) k-means (b) k-medoids
3. Classification Rule Mining: - Decision tree generation using (a) gini index (b) entropy value.

Genetic algorithms were applied to association and classification techniques. Further, k-means and Density Based Spatial Clustering of Applications of Noise (DBSCAN) clustering techniques [1] were applied to a microarray dataset and compared. The microarray dataset was downloaded from internet using the Gene Array Analyzer Software(GAAS) .The clustering was done on the basis of the signal color intensity of the genes in the microarray experiment.

The following results were obtained:-

1. Association: - For smaller databases, the a priori algorithm works better than partition algorithm and for larger databases partition works better.
2. Clustering: - With respect to the number of interchanges, k-medoids algorithm works better than k-means algorithm.
3. Classification: - The results were similar for both the indices (gini index and entropy value).

The application of genetic algorithm improved the efficiency of the association and classification techniques.

For the microarray dataset, it was found that DBSCAN is less efficient than k-means when the database is small but for larger database DBSCAN is more accurate and efficient in terms of no. of clusters and time of execution. DBSCAN execution time increases linearly with the increase in database and was much lesser than that of k-means for larger database.

Owing to the involvement of large datasets and the need to derive results from them, data mining techniques can be effectively put in use in the field of Bio-informatics [2]. The techniques can be applied to find associations among the genes, cluster similar gene and protein sequences and draw decision trees to classify the genes. Further, the data mining techniques can be made more efficient by applying genetic algorithms which greatly improves the search procedure and reduces the execution time.

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 INTRODUCTION TO DATA MINING

With the enormous amount of data stored in files, databases, and other repositories, it is increasingly important, if not necessary, to develop powerful means for analysis and interpretation of data and for the extraction of interesting knowledge that could help in decision-making. Data Mining, also popularly known as Knowledge Discovery in Databases (KDD)[4], refers to as ”the nontrivial process of identifying valid, novel, potentially useful and ultimately understandable pattern in data”. The following figure (Figure 1.1) shows data mining as a step in an iterative knowledge discovery process.

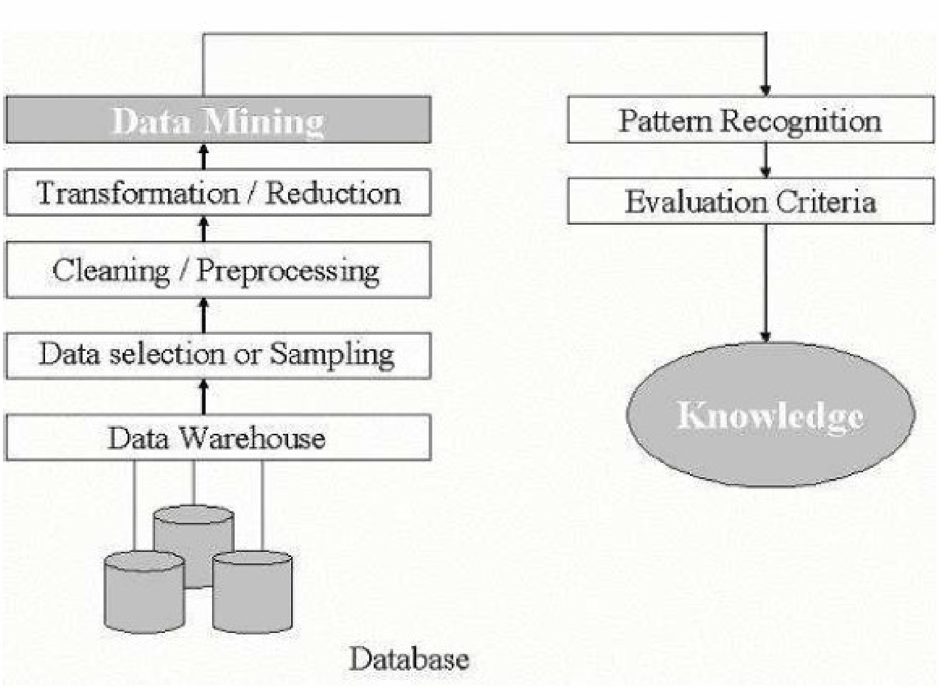

Fig 1.1: An Overview of the Steps Comprising the KDD Process

The iterative process[1] consists of the following steps:

1. Data cleaning**:** Also known as data cleansing, it is a phase in which noisy data and irrelevant data are removed from the collection.

2. Data integration**:** At this stage, multiple data sources, often heterogeneous, may be combined in a common source.

3. Data selection**:** At this step, the data relevant to the analysis is decided on and retrieved from the data collection.

4. Data mining**:** It is the crucial step in which clever techniques are applied to extract data patterns potentially useful.

5. Pattern evaluation**:** In this step, strictly interesting patterns representing knowledge are identified based on given measures.

6. Knowledge representation**:** It is the final phase in which the discovered knowledge is visually represented to the user. This essential step uses visualization techniques to help users understand and interpret the data mining results.

The KDD is an iterative process. Once the discovered knowledge is presented to the user, the evaluation measures can be enhanced, the mining can be further refined, new data can be selected or further transformed, or new data sources can be integrated, in order to get different, more appropriate results.

1.2 INTRODUCTION TO BIOINFORMATICS

Bioinformatics involves the manipulation, searching and data mining of DNA sequence data. The development of techniques to store and search DNA sequences[18] have led to widely-applied advances in computer science, especially string searching algorithms, machine learning and database theory.

In other applications such as text editors, even simple algorithms for this problem usually suffice, but DNA sequences cause these algorithms to exhibit near-worst case behavior due to their small number of distinct characters. Data sets representing entire genomes' worth of DNA sequences, such as those produced by the Human Genome Project[19], are difficult to use without annotations, which label the locations of genes and regulatory elements on each chromosome.

Regions of DNA sequence that have the characteristic patterns associated with protein or RNA coding genes can be identified by gene finding algorithms[22], which allow researchers to predict the presence of particular gene products in an organism even before they have been isolated experimentally.

Chapter 2

DATA MINING TECHNIQUES

DATA MINING TECHNIQUES

Researchers identify two fundamental goals of data mining : prediction and description. Prediction makes use of existing variables in the database in order to predict future values of interest and description focuses on finding patterns describing the data and the subsequent presentation for user interpretation. The relative emphasis of both prediction and description differ with respect to underlying application and the technique. There are several data mining categories [5] fulfilling these objectives : association rule mining, clustering, classification mining using the techniques such as decision tree, genetic algorithms, machine learning and neural networks. The following data mining categories were studied and analyzed.

2.1 ASSOCIATION

Association rule mining [6] searches interesting correlations among items in a given data set. Let us consider a database of sales transactions; it is desirable here to discover the important associations among items such that the presence of some items in a transaction will imply the presence of other items in the same transaction.

Let I = $\{i_1, i_2, \dots , i_m\}$ be a set of literals, called items. Let D be a set of transactions where each transaction T is a set of items such that $T \subseteq I$. Let X is a set of items. A transaction T is said to contain X if and only if $X \subseteq T$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$ and $X \cap Y = \Phi$.

The rule $X \Rightarrow Y$ holds in the transaction set D with confidence c if c% of transactions in D that contain X also contain Y. The rule $X \Rightarrow Y$ has support s in the transaction set D if s% of transactions in D contains $X \cup Y$.

The task of mining association rules is essentially to discover strong association rules in large databases. According to Han and Kamber [3], the problem of mining association rules is decomposed into the following two sub-problems:

1. Find the item sets that have transaction support above a pre-determined minimum support. These itemsets are called frequent itemsets.
2. Use the large item sets to generate the desired association rules for the database.

After the large item sets are identified, the corresponding association rules can be derived in a straightforward manner. Much of the research work has been focused on the first subproblem and as the database is accessed in this part of the computation, we concentrate on this subproblem. We therefore stress on two association algorithms: - a priori and partition algorithms and then compare the two algorithms.

2.1.1 A Priori Algorithm

The a priori algorithm is also called the level-wise algorithm[7]. This technique uses the property that any subset of a large itemset must be a large itemset. The a priori generates the candidate itemsets by joining the large itemsets of the previous pass and deleting those subsets which are small in the previous pass without considering the transactions in the database.

In the first pass, the itemsets with only one item are counted. The discovered large itemsets of the first pass are used to generate the candidate sets of the second pass. Once the candidate itemsets are found, their supports are counted to discover the large itemsets of size two by scanning the database. In the third pass, the large itemsets of the second pass are considered as the candidate sets to discover large itemsets of this pass. This iterative process terminates when no new large itemsets are found. The notations used in the algorithm are :-

1. L_k: - Set of large k-itemsets (those with minimun support).
2. C_k: - Set of candidate k-itemsets (potentially large itemsets)

The algorithm consists of two phases :- The candidate generation step and pruning step.

1.Candidate generation step (gen_cand_itemsets)

```
$C_k=\Phi$
for all itemsets $l_1 \in L_{k-1}$ do
        for all itemsets $l_2 \in L_{k-1}$ do
                if $l_1[1]=l_2[1]$ ^ $l_1[2]=l_2[2]$ ^...........^ $l_1[k-1]=l_2[k-1]$
                then c= $l_1[1], l_1[2],\ldots, l_1[k-1], l_2[k-1]$
                $C_k= C_k \upsilon \{c\}$
```

2.Pruning step (Prune)

```
For all c $\in C_k$
For all (k-1) subsets d of c do
        If d not belongs to $L_{k-1}$
        Then $C_k = C_k \setminus \{c\}$
```

A Priori Algorithm

```
Initialize: k:=1; $C_1$= all the 1-itemsets;
$L_1$:= {frequent 1-itemsets in D};
while $L_{k-1} \neq \Phi$ do
begin
```

```
        $C_k$ := gen_cand_itemsets with the given $L_{k-1}$
        Prune($C_k$)
        for each transaction T in the database do
                increment the count of all candidate itemsets in $C_k$ that are contained in T;
        $L_k$:= candidates in $C_k$ with minimum support;
        k := k+1;
end
return $\upsilon_k$ $L_k$;
```

2.1.2 Partitioning Algorithm

The partition algorithm is based on the observation that the frequent sets are normally very few in number compared. It reduces the number of database scans to 2. It divides the database into small partitions such that each partition can be handled in the main memory. Let the non-overlapping partitions of the database be D1, D2, ..., Dp. In the first scan, it finds the locally frequent itemsets in each partition Di (1≤i≤p), i.e. {X |X.count ≥ s × |Di|}. The local frequent itemsets, Li, can be found by using a level-wise algorithm such as a priori. Since each partition can fit in the main memory, there will be no additional disk I/O for each partition after loading the partition[8] into the main memory. In the second scan, it uses the property that a frequent itemset in the whole database must be locally frequent in at least one partition of the database. Then the unions of the locally frequent itemsets found in each partition are used as the candidates and are counted through the whole database to find all the frequent itemsets.

Partition favors a homogeneous data distribution. That is, if the count of an itemset is evenly distributed in each partition, then most of the itemsets to be counted in the second scan will be large. However, for a skewed data distribution, most of the itemsets in the second scan may turn out to be small, thus wasting a lot of CPU time counting false itemsets.

Algorithm

```
P = partition_database(T); n=number of partitions;
//Phase 1
        for i=1 to n do begin
        read_in_partition ($T_i$ in P)
        $L^i$ =generate all frequent itemsets of $T_i$ using a priori method in main memory
        end
//Merge Phase
```

```
    for(k=2;$L_k^i \neq \Phi$,i=1,2,3,….n;k++) do begin
            $C_k^G = \upsilon L_i^k$
    end
//Phase 2
    for i=1 to n do begin
            read_in_partition ($T_i$ in P)
            for all candidates $c \in C^G$ compute $s(c)_{Ti}$
    end
    $L^G = \{c \in C^G \mid s(c)_{Ti} >= \sigma)$
Answer = $L^G$
```

2.1.3 Comparison between a priori and partition algorithms

The a priori and partition algorithms were compared using a transaction database. The database represents the sale of 9 items i.e. a1, a2, …, a9 and the rows are the transactions. For eg the first transaction shows that the sale of item 1, item 4 and item 5 have occurred together.

a1	a2	a3	a4	a5	a6	a7	a8	a9
1	0	0	1	1	0	0	0	0
0	1	1	0	1	1	0	1	1
1	0	1	0	1	0	0	0	1
1	1	0	1	1	0	0	0	1
0	0	1	1	0	1	1	0	1
1	0	1	1	1	0	0	1	0
0	0	0	0	1	1	1	0	1
0	1	0	1	1	1	0	0	1
1	1	1	1	0	1	0	0	1
1	0	1	0	0	0	1	0	0

Table No. 2.1: Transaction database

The following results were obtained:-

1. Number of candidate sets generated: (i) A priori - 47
   (ii) Partition - 88
2. Number of database scans: (i) A priori - 3
   (ii) Partition – 2

The experimental results show that a priori works better than partition algorithm for smaller databases while partition works better for larger databases. Partition requires less I/O overhead [9] than a priori because it requires lesser database scans.

2.2 CLUSTERING

Clustering is a technique [10] for the purpose of division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups. An example of clustering is depicted in Figure 2.1. Data modeling puts clustering in a historical perspective rooted in mathematics, statistics, and numerical analysis. From a machine learning perspective clusters correspond to hidden patterns, the search for clusters is unsupervised learning, and the resulting system represents a data concept. Data mining deals with large databases that impose on clustering analysis additional severe computational requirements. These challenges led to the emergence of powerful, broadly applicable data mining clustering methods surveyed below.

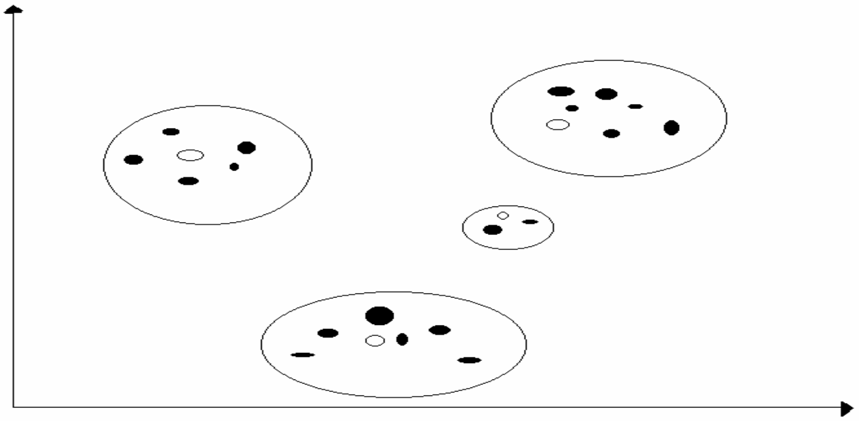

Fig 2.1: Clustering example

The white circle shows different seeds in clusters, Black circle shows valid data points. Data points belonging to the same clusters are shown by a dotted circle. Outliers may also exist in clustering result. Outliers are data points which are not included in any cluster.

2.2.1 K-medoids Algorithm

Under the k-medoids algorithms we use the PAM(Partitioning Around Algorithm)[11].PAM arbitrarily selects k objects from data as medoids. Each of these k objects are representative of k classes. Other objects in the database are classified on the basis of their distances to these k medoids. The algorithm starts with arbitrarily selected k medoids. And iteratively improves upon this selection. In each step, a swap between a selected object Oi and a non-selected object Oj is made as long as such a swap results in an improvement in clustering. To calculate the effect of such a swapping between Ci and Ch, a cost Cih is calculated which is related to the quality of partitioning the non-selected objects to k-clusters represented by the medoids. The algorithm has two important modules: - the partitioning of the database for a given set of medoids and the iterative selection of medoids[1].

Algorithm

```
Input: - Database of objects D
Select arbitrarily k representative objects only from the dataset $K_{med}$.
Mark these objects as selected and the remaining as non-selected
    do for all selected objects $O_i$
        do for all non-selected objects $O_j$
            compute $C_{ih}$
        end do
    end do
select $i_{min},h_{min}$ such that $C_{imin,hmin}= Min_{i.h}\ C_{ih}$
    if $C_{imin,hmin} <0$
        then swap;mark $O_i$ as non-selected and $O_h$ as selected
    repeat
find clusters $C_1, C_2, C_{3.....,}C_k$
```

2.2.2 K-means Algorithm

It is similar to the k-medoids algorithm except that the centroids are computed as the arithmetic mean of all points of a cluster and the cluster centers are not necessarily objects in the database.

Algorithm

```
Input: - Database of objects D
Select arbitrarily k representative objects $K_{mean}$
Mark these objects as selected and the remaining as non-selected
    do for all selected objects $O_i$
        do for all non-selected objects $O_j$
            compute $C_{ih}$
        end do
    end do
select $i_{min},h_{min}$ such that $C_{imin,hmin}= Min_{i.h}\ C_{ih}$
```

```
    if $C_{imin,hmin}$ <0
        then swap;mark $O_i$ as non-selected and $O_h$ as selected
    repeat
find clusters $C_1$, $C_2$, $C_{3.....}$,$C_k$
```

2.2.3 Comparison between k-means and k-medoids algorithm

The k-means and k-medoids algorithm was compared using a student database.

roll	math_mark	phy_mark	chem_mark	avg
1	70	65	75	70
2	89	69	77	78
3	77	86	88	84
4	86	77	67	77
5	94	92	85	90
6	67	66	70	68
7	55	71	65	64
8	76	88	80	81
9	88	68	77	78
10	60	54	68	61

Table No. 2.2 Student database

The following observations were made:-

Number of interchanges: (i) K-means – 116

(ii) K-medoids – 171

It was found that K-means works better than K-medoids since it performs clustering with less number of interchanges.

2.3 CLASSIFICATION USING DECISION TREES

Decision trees, also known as classification trees [12] , are a statistical tool that partitions a set of records into disjunctive classes. The records are given as tuples with several numeric and categorical attributes with one additional attribute being the class to predict.

In general the tree is binary. Each node is either a leaf or has two children. If the node is a leaf then it corresponds to one class and the path from the root gives a complete classification criterion. If the node is not a leaf then the node contains a so-called split criterion that decides where to divide observations based on one of the numeric attributes.

This split criterion is generally an inequality of the form $a_i \leq v_{j,}$ where a_i is some numeric attribute and v_j is specific value. Then each branch corresponds to a yes or no decision to the validity of the inequality. This is why classification trees are also called decision trees. Decision trees algorithm differs in selection of variables to split and how they pick the splitting point.

Construction of the decision tree

The construction of the decision tree involves the following three phases: -

1. Construction phase: - The initial decision tree [13] is constructed in this phase, based on the entire training data set. It requires recursively partitioning the entire data set into two or more sub-partitions using a splitting criteria until a stopping criteria is met.
2. Pruning phase: - The tree constructed in the previous phase may not deliver the best possible set of rules due to over-fitting, the pruning phase removes some of the lower branches and nodes to improve its performance.
3. Processing the pruned tree to improve the understandability.

We used the Classification and Regression Tree (CART) technique to construct the decision tree using (a) gini index and (b) entropy value as splitting index.

2.3.1 Classification and Regression Tree (CART)

CART is one of the more popular methods of constructing the decision tree. It builds a binary decision tree by splitting the records at each node according to a function of a single attribute. The initial split produces two nodes, each of which we now attempt to split in the same manner as the root node. Once again we examine all the input fields to find the splitting candidate. If no split can be found that significantly decreases the diversity of a given node, then we label the node as a leaf node. Eventually only the leaf nodes remain and we have grown a full decision tree. At the end of the tree-growing phase, every record of the training set has been assigned to some leaf of the full decision tree. Each leaf can now be assigned a class and error rate.

We constructed the decision tree using the CART method using

(a) gini index(g_i) as the splitting index

where $g_i = 1 - \sum p_i^2$

(b) entropy value(e_i) as the splitting index

where $e_i = p_i \log(p_i)$

p_i is the frequency of occurrence of the class i in T

2.3.2 Comparison between gini index and entropy value as the splitting index

outlook	temp	Humidity	windy	class
sunny	77	75	true	play
rain	65	88	false	no play
sunny	62	80	true	play
overcast	71	78	true	play
rain	59	89	false	no play
overcast	66	82	false	no play
sunny	61	74	true	play
overcast	75	86	false	no play
sunny	59	76	false	no play
rain	80	88	true	play

Table No. 2.3 Classification database

We used the above table to construct a decision tree using gini index and entropy value as the splitting index and got the following results.

Gini index for outlook = 0.49972892
humidity = 0.49980438
windy = 0.49986857
temp = 0.49997646

Time for Execution:-859 ms

Entropy value for outlook = 1.9997959
humidity = 1.9999048
windy = 1.9999974
temp = 2.000153

Time for Execution:-875 ms

The best splitting attribute in both the indices was found to be outlook and also the order of splitting attributes were found to be same for both the indices.

2.4 SOFT COMPUTING TECHNIQUES

Soft Computing refers to a collection of computational techniques in computer science, artificial intelligence, machine learning and some engineering disciplines, which attempt to study, model and analyze very complex phenomena. Earlier computational approaches could model and precisely analyze only relatively simple systems. More complex systems arising in biology, medicine, the humanities, management sciences, and similar fields often remained intractable to conventional mathematical and analytical methods. This is where soft computing provides the solution.

Key areas of soft computing[2] include the following::

1. Fuzzy logic

Fuzzy logic is derived from fuzzy set theory dealing with reasoning that is approximate rather than precisely deduced from classical predicate logic. It can be thought of as the application side of fuzzy set theory dealing with well thought out real world expert values for a complex problem. Fuzzy logic can be used to control household appliances such as washing machines (which sense load size and detergent concentration and adjust their wash cycles accordingly) and refrigerators.

2. Neural Networks

Artificial neural networks are computer models built to emulate the human pattern recognition function through a similar parallel processing structure of multiple inputs. A neural network consists of a set of fundamental processing elements (also called neurons) that are distributed in a few hierarchical layers. Most neural networks contain three types of layers: input, hidden, and output. After each neuron in a hidden layer receives the inputs from all of the neurons in a layer ahead of it (typically an input layer), the values are added through applied weights and converted to an output value by an activation function (e.g., the Sigmoid function). Then, the output is passed to all of the neurons in the next layer, providing a feed-forward path to the output layer.

3. Statistical Inferences

Statistics provides a solid theoretical foundation for the problem of data analysis. Through hypothesis validation and/or exploratory data analysis, statistical techniques give asymptotic results that can be used to describe the likelihood in large samples. The basic statistical exploratory methods include such techniques as examining distribution of variables, reviewing large correlation matrices for coefficients that meet certain thresholds, and examining multidimensional frequency tables.

4. Rule induction

Induction models belong to the logical, pattern distillation based approaches of data mining. Based on data sets, these techniques produce a set of if-then rules to represent significant patterns and create prediction models. Such models are fully transparent and provide complete explanations of their predictions**.**

5. Genetic Algorithm

They are search algorithms based on the mechanics of natural genetics i.e. operations existing in nature. They combine a Darwinian 'survival of the fittest' approach with a structured, yet randomized, information exchange. The advantage [14] is that they can search complex and large amount of spaces efficiently and locate near optimal solutions rapidly.

The algorithm operates through a simple cycle as shown in fig 2.2:-

1. Creation of a population of strings
2. Evaluation of each string
3. Selection of the best strings
4. Genetic manipulation to create a new population of strings.

The GA maps strings of numbers to each potential solution. Each solution becomes an individual in the population and each string becomes a representation of an individual.

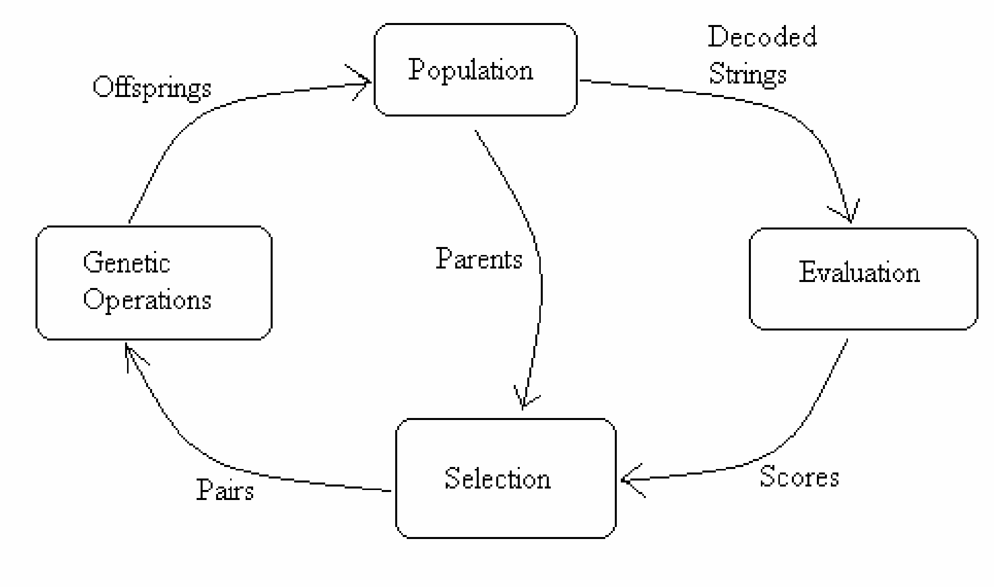

Fig: 2.2: The Genetic Algorithm cycle

Two operators are used : Crossover and Mutation. The offspring generated by this process replaces the older population and the cycle is repeated until a desired level of fitness is achieved [15].

Crossover is one of the genetic operators used to recombine the population's genetic material. It takes two chromosomes and swaps part of their genetic information to produce new chromosomes. The mutation operator introduces new genetic structures in the population by randomly changing some of its building blocks.

Genetic algorithms were applied on the data mining techniques to improve their search procedure and reduce programming complexity. GA was applied on the association and classification rule mining techniques [16].

2.4.1 Application of Genetic algorithm into Association rule mining

GA was applied to association rule mining in the following manner

Algorithm Genetic

```
Input: Database that was taken for association rule mining
Read the initially provided random rules that are of the form ‘100100100’ which implies out
of 9 items, item 1, item 4 and item 7 occur together in the database.
do while consecutive generations are equal
    call function Means()          // to determine the meaning of the representation.
    call function Fit()            //determine fitness value i.e. number of times the rule
                                   //occurs in the database
    call function Reproduction()   // select most fit rules and make their copies
    call function Crossover()      // cross over the rules to create new rules
    call function Means ()
    call function fit()
end while
```

2.4.2 Application of Genetic algorithm into Classification rules

GA was applied to classification rules [17] in the following manner

Algorithm Genetic

```
Input: Database that was taken for classification rules
Read the initially provided random rules that are of the form ‘1001001001’
Function Means ()
It determines the values of various parameters for each of the gene value.
First three bits represent the three possible values for outlook: sunny, overcast or rainy. Next two bits represent the temperature: either greater or lesser than user specified value. . Next two bits represent the humidity: either greater or lesser than user specified value. Next two bits represent the wind conditions: either true or false. Final bit represents the class label: either play or no play.
do while consecutive generations are equal
    call function Means()          // to determine the meaning of the representation.
    call function Fit()            //determine fitness value i.e. number of times the rule
                                   //occurs in the database
    call function Reproduction()   // select most fit rules and make their copies
    call function Crossover()      // cross over the rules to create new rules
```

```
        call function Means ()
        call function fit()
end while
```

Results

Genetic algorithm was successfully applied to classification and association rule mining and the decision tree rules and maximal frequent set found out were more appropriate. The results were obtained in lesser time .

Chapter 3

STUDY ON BIOINFORMATICS

3.1 FUNDAMENTALS OF MOLECULAR BIOLOGY

Deoxyribonucleic acid (DNA) is the basis of genetic material. In other words, the information stored in DNA allows the organization of inanimate molecules into functioning in living cells and organisms. These groupings can regulate their internal chemical composition, growth and reproduction. The various units that govern these characteristics, be it chemical composition or nose size, are called genes. Genes themselves contain their information as a specific sequence of nucleotides that are found in DNA molecules. Only four different nucleotides (or bases) are used in DNA molecules: adenine, guanine, cytosine and thymine (A, G, C and T). All the information within each gene comes simply from the order in which those nucleotides are found. Complicated genes can be many thousands of nucleotides long. Many genes code for proteins and best estimates are that it takes several tens of thousands of different proteins to make a human being. Fundamentally, a protein is a long chain (a polymer) of building blocks called amino acids. Varying combinations of only 20 different amino acids are used to build all of the proteins in a human being.

3.2 CENTRAL DOGMA OF MOLECULAR BIOLOGY

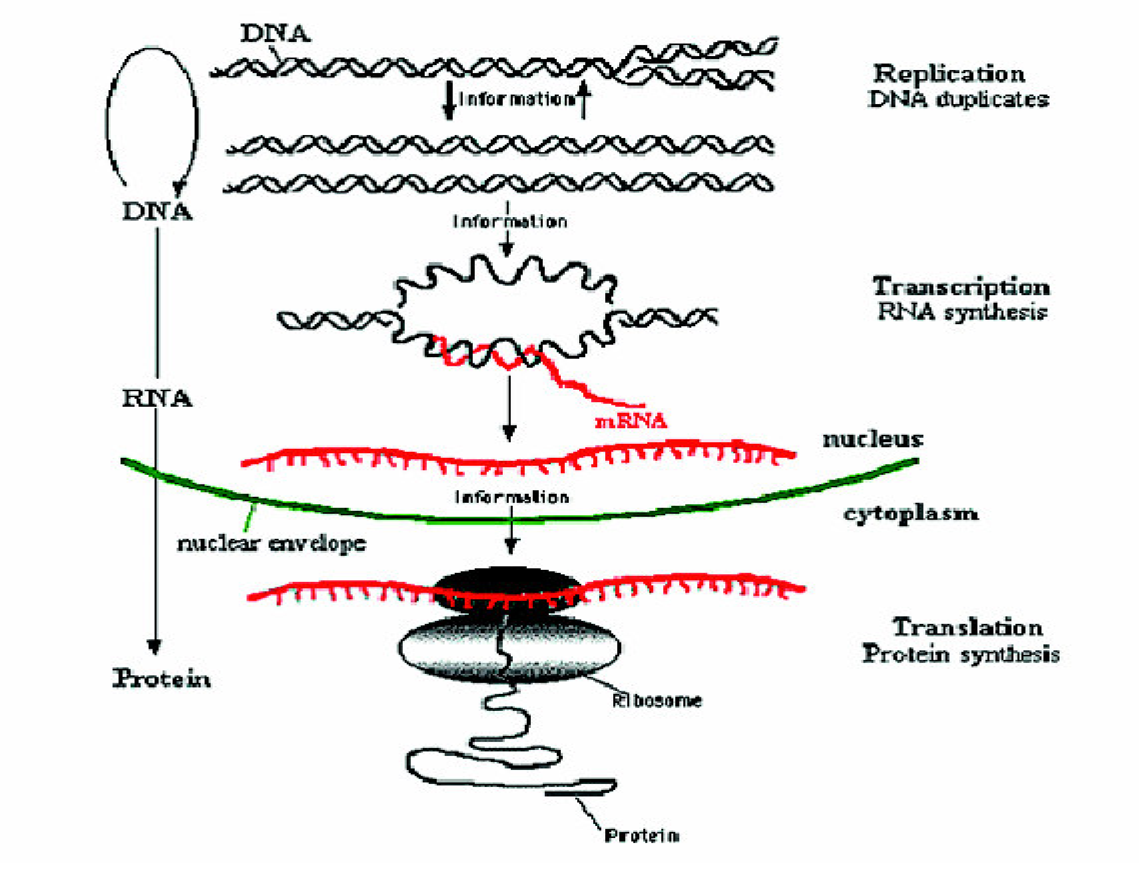

Fig 3.1 The Central Dogma of Molecular Biology

The simplified version of the central dogma is as shown in the diagram. It consists of the following steps: Replication of DNA, Transcription of DNA to RNA, Translation to proteins and protein folding.

Replication

In the process of DNA replication, all the information in the double-stranded sequence of a DNA helix is duplicated on each strand. This reversible and specific interaction between complementary base pairs is critical for all the functions of DNA in living organisms

Transcription

In this process, the DNA is transcribed to single-stranded nuclear RNA (nRNA) which is then processed to form mature messenger RNA (mRNA).Small nuclear RNA(snRNA) is involved in the maturation process, which includes excising the introns (non-coding segments) and concatenating the remaining exons (coding sequences) according to their original order in the mRNA. The RNA differs from DNA in that it contains a nitrogenous base Uracil(U) instead of the Thymine(T). mRNA is transported through the nuclear membrane to the cytoplasm where the translation of mRNA to protein occurs with the aid of ribosomes.

Translation

The translation of mRNA to protein is facilitated by transfer RNA (tRNA), which associates with the 20 common amino acids and controls the sequential binding of the amino acids. The amino acids are added to the growing protein sequence one at a time as the ribosome moves from codon to codon along the mRNA. At the stop codon, the translation ends and the protein is released by the ribosome.

Protein folding

Before the protein is transported outside the cell to perform or promote a variety of tasks, it is usually modified by adding a sugar, for example, and it takes on a characteristic folded three-dimensional form forming polypeptide sequences.

3.3 MICROARRAY EXPERIMENTAL ANALYSIS

Molecular Biology research evolves through the development of the technologies used for carrying them out. It is not possible to research on a large number of genes using traditional methods. One can analyze the expression of many genes in a single reaction quickly and in an efficient manner using microarrays. Microarray technology has empowered the scientific community to understand the fundamental aspects underlining the growth and development of life as well as to explore the genetic causes of anomalies occurring in the functioning of the human body.

All the data is collected and a profile is generated for gene expression in the cell.

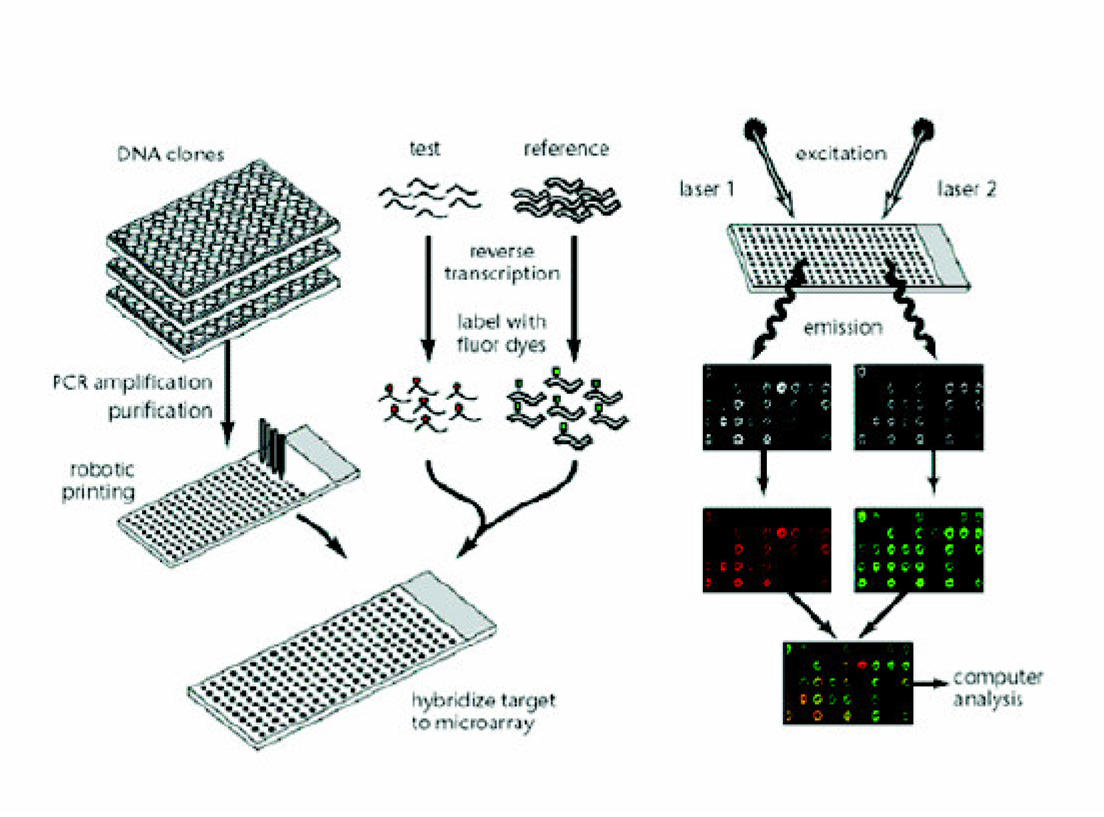

Fig 3.2: Microarray experimental analysis

DNA microarrays [20] are created by robotic machines that arrange minuscule amounts of hundreds or thousands of gene sequences on a single microscope slide. Researchers have a database of over 40,000 gene sequences that they can use for this purpose. When a gene is activated, cellular machinery begins to copy certain segments of that gene. The mRNA produced by the cells bind to the original portion of the DNA strand from which it was copied. To determine which genes are turned on and which are turned off in a given cell, a researcher must first collect the mRNA molecules present in that cell. The researcher then labels each mRNA molecule by attaching a fluorescent dye. Next, the researcher places the labeled mRNA onto a DNA microarray slide. The messenger RNA that was present in the cell will then hybridize or bind to its complementary DNA on the microarray, leaving its fluorescent tag. A researcher must then use a special scanner to measure the fluorescent areas on the microarray. Researchers frequently use this technique to examine the activity of various genes at different times. This activity is referred to as the gene expression value of the genes.

Another application of microarrays is if we want to know the activity of genes that are responsible for a disease under different conditions. In this experimental setup, the cDNA derived from the mRNA of known genes is immobilized. The sample has genes from both the normal as well as the diseased tissues. This expression pattern is then compared to the expression pattern of a gene responsible for a disease. Spots with more intensity are obtained for diseased tissue gene if the gene is over expressed in the diseased condition.

Chapter 4

REQUIREMENT OF DATA MINING TECHNIQUES IN BIOINFORMATICS

4.1 NEED OF DATA MINING IN BIOINFORMATICS

The entire human genome, the complete set of genetic information within each human cell has now been determined. Understanding these genetic instructions promises to allow scientists to better understand the nature of diseases and their cures, to identify the mechanisms underlying biological processes such as growth and ageing and to clearly track our evolution and its relationship with other species. The key obstacle lying between investigators and the knowledge they seek is the sheer volume of data available. This is evident from the following figure which shows the rapid increase in the number of base pairs and DNA sequences in the repository of GenBank.

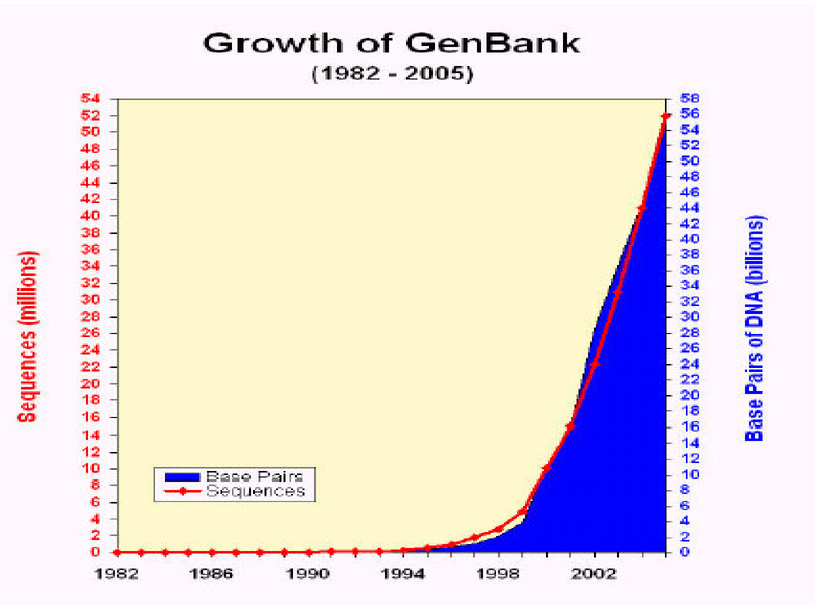

Fig 4.1: Growth of GenBank

Biologists, like most natural scientists, are trained primarily to gather new information. Until recently, biology lacked the tools to analyze massive repositories of information such as the human genome database. Luckily, the discipline of computer science has been developing methods and approaches well suited to help biologists manage and analyze the incredible amounts of data that promise to profoundly improve the human condition. Data mining is one such technology.

4.2 APPLICATION OF CLUSTERING TECHNIQUE TO MICROARRAY ANALYSIS

The application procedure consists of the following steps:-

Inspect the micro array and qualitatively assign each spot on the micro array a value corresponding to its gene expression level, which is taken directly from its color intensity. To analyze gene expression patterns [21] using a clustering algorithm, we do the following

(1) We construct a similarity pattern for the different genes

(2) Cluster the genes based on their similarity using a clustering algorithm

(3) We end up with clusters of similar genes, grouped by their expression levels

We have used two clustering algorithms: - k-means and DBSCAN.

4.2.1 Density Based Spatial Clustering of Applications of Noise (DBSCAN)

Density Based Spatial Clustering of Applications of Noise (DBSCAN) uses a density based notion of clustering of arbitrary shapes. For each object of a cluster, the neighborhood of a given radius must contain atleast a minimum number of data objects. Such objects are called core objects. The algorithm gradually converts the unclassified objects into classified or noise objects.

Algorithm

```
DBSCAN Algorithm ( D, Є, Minpts)
Input: Database of objects D
do for all O Є D
    if O is unclassified
        call function Expand_cluster(O, Є, Minpts)
end do
function Expand_Cluster(O, D, Є, Minpts)
    get the Є-neighbourhood of O as $N_{Є}(O)$
    if  $| N_{Є}(O)| < Minpts$
        mark O as Noise
    return
    else
        select a cluster_id & mark all objects of $N_{Є}(O)$ with this id and put them into
        candidate objects
        do while candidate object is not empty
            select an object from candidate objects as curr_obj
            delete curr_obj from candidate objects
            retrieve $N_{Є}(curr\_obj)$
            if $| N_{Є}(O)| \geq Minpts$
                select all objects in $N_{Є}(curr\_obj)$ not yet classified or marked
                as noise
                mark all of the objects with cluster_id
            include the unclassified objects into candidate objects
    end do
return
```

4.2.2 Comparison of results of k-means and DBSCAN

Location	MSI	MBI	index
10102	1.162	1.06	1
10201	0922	0.994	2
10202	1.056	1.076	3
10301	0.97	0.983	4
10302	1.047	1.05	5
10401	0.979	0.97	6
10402	1.03	1.068	7
10501	0.959	1.027	8
10502	1.029	1.113	9
10601	1.004	1.045	10

Table No 4.1 : Microarray Database

The two clustering techniques, DBSCAN and K-means were applied on the microarray dataset. The dataset was procured using the Gene Array Analyzer Software (GAAS). Location signifies the position of the gene in the microarray chip. MSI (Mean Signal Intensity) is the intensity of the dye that was present in the location. MBI (Mean Background Intensity) is the intensity of the less illuminated location. The clustering were done on the basis of MSI. The following results were obtained.

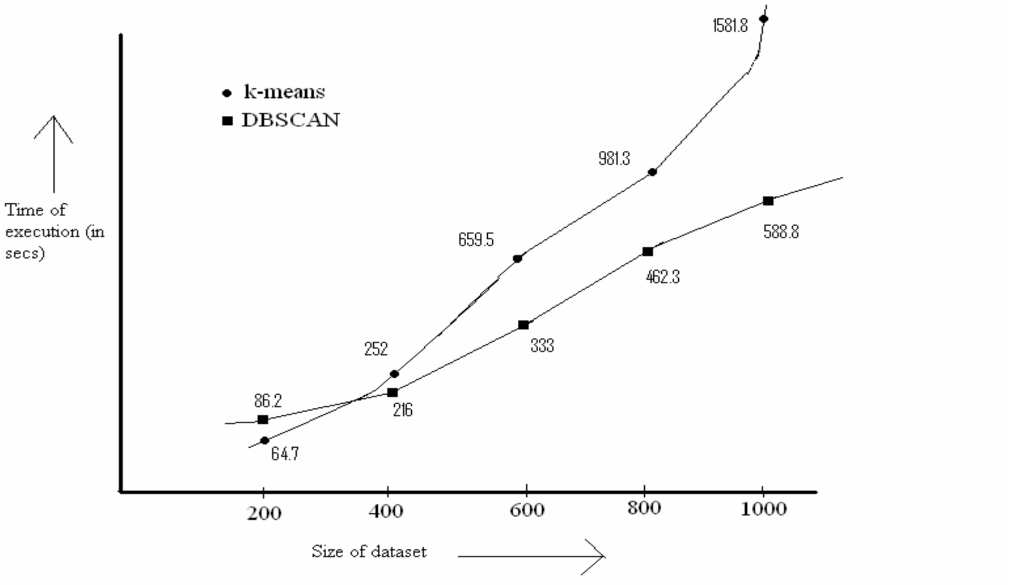

Fig 4.2: Performance comparison of K-means and DBSCAN based on time of execution

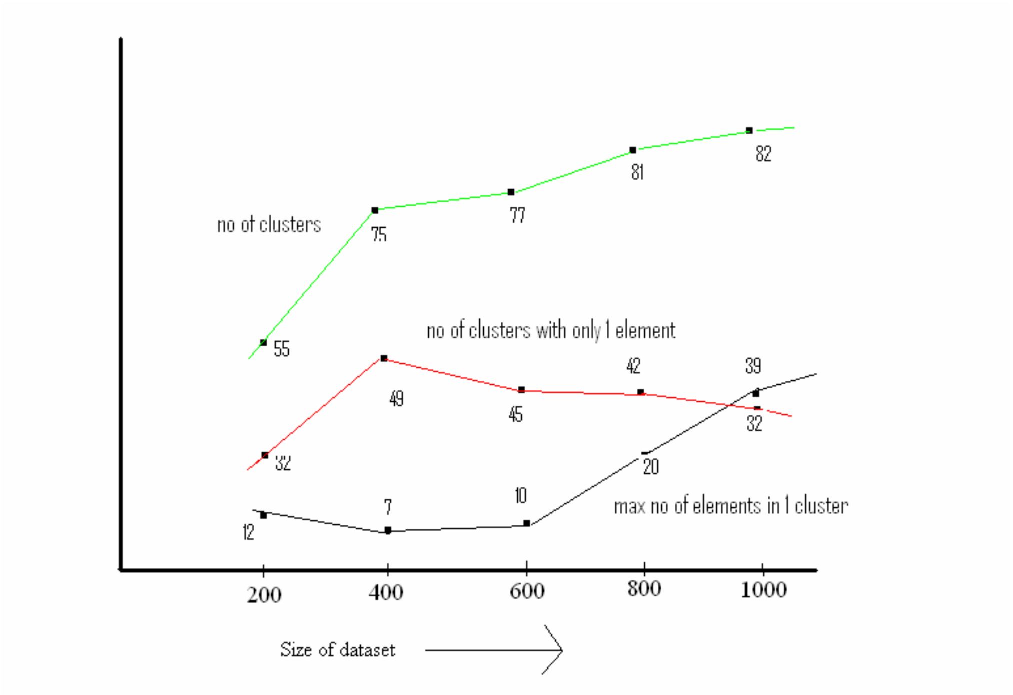

Fig 4.3: Performance of DBSCAN with increasing database size.

For the microarray dataset, it was found that DBSCAN is less efficient than k-means when the database is small but for larger database DBSCAN is more accurate and efficient in terms of no. of clusters and time of execution. DBSCAN execution time increases linearly with the increase in database and was much lesser than that of k-means for larger database.

Chapter 6

CONCLUSION

CONCLUSION

In this report an in-depth study of the varied data mining techniques was made. It was shown how genetic algorithms can be used to optimize the data mining algorithms. The report then gives an introduction to molecular biology and bioinformatics. Then the microarray experimental analysis was studied and the clustering techniques were applied to mine the microarray data. The report basically underlines the role of clustering analysis to cluster genes into groups of similar character. The microarray experiment produces thousands of samples for each gene, clustering can be effectively used to group these genes into disease causing genes and normal genes and to study the various characteristics of different genes under different conditions. This can be used for drug treatment considering the response of the genes to drugs paving the way for diagnosis of incurable diseases like AIDS, Alzheimer's disease. It can also be used to identify the mechanisms underlying biological processes such as growth and ageing and to track the process of our evolution. Data mining is therefore an effective technique to solve the problem of enormous data faced by researchers in their quest to solve the puzzles of our life.

REFERENCES

[1] Pujari, Arun . *Data Mining Techniques*. Nancy: Universities Press, 2001.

[2] Zhang , Dongsang and Zhou, Lina. "Data Mining Techniques in Financial Application". *IEEE Transactions on Systems, Man and Cybernetics – Part C : Applications and Reviews,* Vol – 34, No- 4, Nov-2004, pp. 513 – 522.

[3] Han, Jiawei and Kamber, Micheline. *Data mining: concepts and techniques*, San Francisco: Morgan Kaufmann Publishers Inc., CA, 2000

[4] Piateski, Gregory and Frawley, William. *Knowledge Discovery in Databases*, Cambridge: MIT Press, MA, 1991

[5] Edder, John F and Abbott , Dean. "A Comparison of Leading Data Mining Tools." *Fourth International Conference on Knowledge discovery and Data Mining*, Friday, Aug 28, 1998, NY, pp 19-25.

[6] Agrawal, R., Imielinski, T. and Swami, A. "Mining associations between sets of items in massive databases." *ACM SIGMOD International Conference on Management of Data*, pp 207--216, Washington, DC, May 1993.

[7] Hipp, Jochen , Guntzer, Ullrich and Nakhaeizadeh, Gholamreza, "Algorithms for Association Rule Mining – A general Survey and Comparison". *SIGKDD explorations*, Vol 2, Issue – 1, pp 58 – 63, Mar – 2004.

[8] Agrawal, Rakesh and Srikant, Ramakrishnan. "Fast Algorithms for Mining Association Rules in Large Databases" , *Proceedings of the 20th International Conference on Very Large Data Bases*, pp.487-499, September 12-15, 1994

[9] Gyorodi, Robert S . "A Comparative Study of Iterative Algorithm in Association Rules Mining", *Studies in Informatics and Control*, Vol – 12, No – 3, pp 205 – 212, Sept 2003.

[10] Jain , A. K. , Murty, M. N. and Flyn, P. J., "Data Clustering : A Review", *ACM Computing Surveys,* Vol – 31, No – 3, pp 264 – 323, 1999.

[11] Salem, S. A. and Nandi, A. K., "New Assessment Criteria for Clustering Algorithms", *IEEE Workshop on Machine Learning for Signal Processing*, Vol – 7803 – 9518, Sept 2005, pp 285 – 290.

[12] Cohen, W., "Fast Effective Rule Induction", *In Proceedings of the 12th International Conference on Machine Learning,* pp 115 – 123, 1995.

[13] Kamber, M., Winstone, L., et al. "Generalization and decision tree induction: efficient classification in data mining". *Proceedings of the 7th International Workshop on Research*

Issues in Data Engineering (RIDE '97) High Performance Database Management for Large-Scale Applications, pp.111, April 07-08, 1997

[14] Goldberg, David E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston: Addison-Wesley Longman Publishing Co., 1989

[15] Punch W. F., Pei M., et al "Further Research on Future Selection and Classification using Genetic Algorithms", *5th International Conference on Genetic Algorithm*, Champaign IL, pp 557 – 564, 1993.

[16] Carvalho, Deborah R. and Frietas, Alex A., "A Hybrid Decision Tree / Genetic Algorithm Method for Data Mining ", *Evolutionary Computation I(2)*, pp 101 – 125, MIT Press, 1993.

[17] Tsuchiya, T., Ishihara, S., Matsubara, Y., Nishiro, T., Magamachi, M, "A Method of Learning Decision Tree Using Genetic Algorithm and its Application to Kansei Engineering System", *IEEE SMC ' 99 Conference Proceedings*, Vol – 6, pp 279 – 283, Oct – 99.

[18] Bergeron, Bryan. *Bioinformatics Computing*. New Delhi: Pearson Education,2003..

[19] Luscombe, N.M., Greenbaum, D. and Gerstein, M : "What is Bioinformatics? A Proposed Definition and Overview of the Field.".*Methods of Information in Medicine*,40(4),pp 346-358,May 2001

[20] Piatetsky-Shapiro, G. and Tamayo, P : "Microarray Data Mining: Facing the Challenges." *SIGKDD Explorations*, 5(2), pp 1-5, 2003 .

[21] Liu, L., Yang, Jiong. and Tung, Anthony. "Data Mining Techniques for Microarray Datasets." *Proceedings of the 21st International Conference on Data 13 Engineering 2005 IEEE*. pp 182-192 ,2005

[22] Shah, Shital C. and Kusiak, Andrew, " Data Mining and Genetic Algorithm Based Gene Selection", *Artificial Intelligence in Medicine 2004*, (31), pp 183-196, Vol – 2139, 2004.