استخلاص قوانين استنتاجية من قواعد البيانات باستخدام البرمجة المرنة

رسالة مقدمة الى
مجلس كلية العلوم – جامعة بابل
وهي جزء من متطلبات نيل درجة ماجستير علوم
في علوم الحاسبات

من

مهدي عبد سلمان

تشرين ثاني- ٢٠٠٥ شوال – ١٤٢٦

Extracting Rules from Databases using Soft Computing

A Thesis
Submitted to the Council of College of Science
University of Babylon
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Computer Science

By
Mahdi A. Salman

November-٢٠٠٥ **Shawwal-١٤٢٦**

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

(فتَعَالَى اللَّهُ الْمَلِكُ الْحَقُّ وَلا تَعْجَلْ بِالْقُرْآنِ مِنْ قَبْلِ أَنْ يُقْضَى إِلَيْكَ وَحْيُهُ وَقُلْ رَبِّ زِدْنِي عِلْمًا)

صدق الله العلي العظيم
طه (١١٤)

Supervisor Certification

I certify that this thesis was prepared under my supervision at the department of Computer Science /College of Science / Babylon University, by **Mahdi A. Salman** as partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Signature:

Name: **Dr. Nabeel H. Kaghed**

Title: **Professor**

Date: / / ٢٠٠٥

In view of the available recommendations, I forward this thesis for debate by the examination committee.

Signature:

Name: **Dr**. **Abbas M. Al-Bakrey**

Title: **Head of Department of Computer Science**

Date: / / ٢٠٠٥

Certification of the Examination Committee

We chairman and members of the examination committee, certify that we have studied the thesis entitled (**Extracting Rules from Databases Using Soft Computing**) presented by the student **Mahdi A. Salman** and examined him in its content and in what is related to it, and we have found it worthy to be accepted for the degree of Master of Science in Computer Science with (**Excellent**) degree.

Signature
Name: **Dr. Abed Al-Hassan Khalawy**
Title: **Prof. Assistant**
Date: /١٢/٢٠٠٥
(Chairman)

Signature
Name: **Dr**. **Abbas M. Al-Bakrey**
Title: **Lecturer**
Date: /١٢/٢٠٠٥
(Member)

Signature
Name: **Dr. Hussain Atia Hatif**
Title: **Lecturer**
Date: /١٢/٢٠٠٥
(Member)

Signature:
Name: **Dr. Nabeel H. Kaghed**
Title: **Professor**
Date**:** /١٢/ ٢٠٠٥
(Supervisor)

Signature
Name: **Dr**. **Oda Mizi'l Yasser Alzamely**
Title: **Professor**
Date**:** /١٢/٢٠٠٥
(Dean of College of Science – Babylon university)

Dedication

To my family
(Father, Mother, brothers, Sisters)

To my friends
(Ayed, Safa , Kamal, Mohammed)

To my department's Staff

Acknowledgment

Its pleasure to acknowledge my debt for many people involved in presenting this thesis: First of all I would like to express my sincere gratitude and appreciation to my supervisor Dr. Nabeel H. Kaghed for his efficient guidance, supervision and untiring efforts during the course of this work. Gratitude and thanks are for my closely friend Muhammad U. Mahdi. Special thanks go to my family for their continuous support and encouragement during the period of my study. Finally, I would like to thank the staff of the department of computer science for the great help they've introduced to me.

Mahdi

ABSTRACT

As the amounts of data stored in databases over the world has grown almost exponentially the last years, the need for efficient methods to extrac knowledge from these databases become more important. Data Mining, also called Knowledge Discovery in Databases (KDD), is a sub field of Machine Learning tries to respond to those needs. Data Mining is not just a single method or a single technique but rather a spectrum of different approaches which searches for patterns and relationships that are hidden among the vast amounts of data.

In this thesis Artificial Neural Network (ANN)-Unsupervised Kohonen was used to cluster (group) the database records. But before doing the clustering, it is important to determine the initial number of clusters and if possible the good initialization of the neural network weights. The Genetic Algorithm (GA) will do this task. When the number of cluster has been determined, the neural network using Winner-Take-All learning rule then will be trained. After training, the network (recall mode) was used to classify the database records. Statistical analysis was used to determine the active attributes for each class. Some attributes in linguistic terms can be expressed through using Fuzzy Logic (FL). Suitable rules that describe classes then formed. The method has the ability to deal with any databases of any domain. It works easily, and fast. The databases used are from various domains: medicine, housing, and cars marketing. VB٦ programming language has been used to build DBRuleExtractor system on P-IV computer.

الخلاصة

نتيجة للنمو الهائل الحاصل للبيانات المخزونة في قواعد بيانات حول العالم في السنوات الاخيرة، اصبحت الحاجة الى اساليب كفوءة لغرض استخلاص المعرفة من قواعد البيانات اكثر اهمية. ان التنجيم عن البيانات ايضا تسمى استكشاف المعرفة في قواعد البيانات، وهو احد فروع تعليم الالة، هو احد المحاولات للاستجابة لهذه الحاجة الملحة. لا يمكن اعتبار مبدا التنجيم عن البيانات على انه مجرد طريقة منفردة او اسلوب منفرد ولكنه طيف من طرق مختلفة تبحث عن النماذج والعلاقات المخفية في كميات هائلة من البيانات.

في هذه الرسالة استخدمت الشبكات العصبية نوع (Unsupervised Kohonen) لاجراء عملية عنقدة لسجلات قاعدة البيانات ولكن قبل اجراء عملية العنقدة من المهم تحديد العدد الابتدائي لعدد العناقيد وان امكن تهية اوزان ابتدائية مناسبة للشبكة العصبية، لقد استخدمت الخوارزميات الجينية لهذا الغرض. عند تحديد العدد الاولي للعناقيد يمكن تدريب الشبكة العصبية باستخدام طريقة تدريب (Winner-Take-All) . بعد اكمال عملية التدريب تصنف السجلات في عملية استدعاء للشبكة (Recall) . وبدراسة كل صنف (مجموعة) احصائيا تحدد اهم الخصائص النشطة لها. ومن خلال المنطق المضبب نعبر عن بعض الصفات بتعابير لغوية. بعدها تشكل مجموعة قواعد استنتاجية تصف كل صنف. ان لهذا الاسلوب قدرة على التعامل مع اي نوع من قواعد البيانات من اي مجال كان. كما انها سريعة وسهلة. كما ان النماذج المكتشفة هي في صورة قواعد استنتاجية. اما قواعد البيانات المستخدمة فهي من مجالات متعددة مثل الطب وعمليات التوطين ومبيعات سيارات. تم استخدام لغة البرمجة VB٦ لبناء نظام استخلاص القواعد وعلى حاسبة بنتيوم P-IV.

Abbreviations

ANFIS	Adaptive-Network-based Fuzzy Inference Systems
ANN	Artificial Neural Networks
DBI	Davies Bouldin Index
DBRulesExtractor	Database Rules Extractor System
DM	Data Mining
FSOM	Fuzzy Self-Organization Map
GA	Genetic Algorithm
GC	Genetic Computing
KDD	Knowledge Discovery in Database
MLP	Multi-Layer Perceptron
MSE	Mean Square Error
NC	Neural Computing
PC	Probabilistic Computing
RBF	Radial Base Function
SC	Soft Computing
SOM	Self-Organization Map

Table of Contents

Chapter One: Introduction.

Chapter Two: Knowledge Discovery in databases, Data mining and Soft Computing.

Chapter Three: Development of DBRuleExtractor System.

CHAPTER ONE

Introduction

CHAPTER TWO

Knowledge Discovery in Databases, Data Mining and Soft Computing.

CHAPTER THREE

Development of DBRulesExtractor System

CHAPTER FOUR

Case Study, Conclusion, and Future Works

REFRENCES

APPENDICES

١.١ General Concepts

The digital revolution has made digitized information easy to capture and fairly inexpensive to store. With the development of computer hardware and software and the rapid computerization of business, huge amount of data have been collected and stored in databases. The rate at which such data is stored is growing at a phenomenal rate. As a result, traditional ad hoc mixtures of statistical techniques and data management tools are no longer adequate for analyzing this vast collection of data [١, ٤٥].

Raw data is rarely of direct benefit. Its true value is predicated on the ability to extract information useful for decision support or exploration, and understanding the phenomenon governing the data source. In most domains, data analysis was traditionally a manual process. One or more analysts would become intimately familiar with the data and, with the help of statistical techniques, provide summaries, and generate reports. In effect, the analyst acted as a sophisticated query processor. However, such an approach rapidly breaks down as the size of data grows and the number of dimensions increases. Databases containing number of data in the order $١٠^{٩}$ and dimension in the order of $١٠^{٣}$ are becoming increasingly common [٤٥]. When the scale of data manipulation, exploration and inferencing goes beyond human capacities, people look to computing technologies for automating the process.

All these have prompted the need for intelligent data analysis methodologies, which could discover useful knowledge from data.

The term KDD refers to the overall process of *knowledge discovery in databases*. ***Data mining*** is a particular step in this process, involving the application of specific algorithms for extracting patterns (models) from data. The additional steps in the KDD process, such as data preparation, data selection, data cleaning, incorporation of appropriate prior knowledge, and proper interpretation of the results of mining, ensures that useful knowledge is derived from the data[٢٤].

The subject of KDD has evolved, and continues to evolve, from the intersection of research from such fields as databases, machine learning, pattern recognition, statistics, artificial intelligence, reasoning with uncertainties, knowledge acquisition for expert systems, data visualization, machine discovery, and high-performance computing. KDD systems incorporate theories, algorithms, and methods from all these fields. Many successful applications have been reported from various sectors such as marketing, finance, banking, manufacturing, and telecommunications. Database theories and tools provide the necessary infrastructure to store, access and manipulate data. *Data warehousing*, a recently popularized term, refers to the current business trends in collecting and cleaning transactional data and making them available for analysis and decision support [٢٤].

Fields concerned with inferring models from data include statistical pattern recognition, applied statistics, machine learning and neural computing. A natural question that arises is: how is KDD different from those fields? KDD focuses on the overall process of knowledge discovery from large volumes of data, including the storage and accessing of such data, scaling of algorithms to massive data sets, interpretation and visualization of results, and the modeling and support of the overall human machine interaction.

Data mining is a form of knowledge discovery essential for solving problems in a specific domain. Individual data sets may be gathered and studied collectively for purposes other than those for which they were originally created. New knowledge may be obtained in the process while eliminating one of the largest costs, viz., data collection. Medical data, for example, often exists in vast quantities in an unstructured format. The application of data mining can facilitate systematic analysis in such cases. Medical data, however, requires a large amount of preprocessing in order to be useful. Here numeric and textual information may be interspersed, different symbols can be used with the same meaning, redundancy often exists in data, erroneous/misspelled medical terms are common, and the data is frequently rather sparse. A robust preprocessing system is required in order to extract any kind of knowledge from even medium-sized medical data sets [٤٤,٤٥].

Soft computing is a consortium of methodologies that works synergistically and provides, in one form or another, flexible information processing capability for handling real-life ambiguous situations. Its aim is to exploit the tolerance for imprecision, uncertainty, approximate reasoning, and partial truth in order to achieve tractability, robustness, and low-cost solutions. The guiding principle is to devise methods of computation that lead to an acceptable solution at low cost by seeking for an approximate solution to an imprecisely/precisely formulated problem [٢٩].

Soft computing methodologies (involving fuzzy sets, neural networks and genetic algorithms) are most widely applied in the data mining. Fuzzy sets provide a natural framework for the process in dealing with uncertainty. Neural networks are widely used for classification and rule generation. Genetic algorithms are involved in various optimization and search processes.

١.٢ Literatures Survey

In ١٩٩٨, T. Nomura; and T. Miyoshi [٤٧] proposed an automatic fuzzy rule extraction method using the hybrid model of the FSOM and the GA with numerical Chromosomes.

In ١٩٩٩ K. McGarry, J. Tait, S. Wermter, and J. MacIntyre,[٣٤] showed that the weights and cluster centers could be directly interpreted as antecedents in a symbolic IF..THEN type rule.

In ٢٠٠٠, P. Mitra and S. Mitra and K. Sankar Pal [٣٩] described a way of designing a hybrid decision support system in soft computing paradigm for detecting the different stages of cervical cancer.

In ٢٠٠١, K. Sankar Pal, S. Mitra and P. Mitra,[٤٤] presented a methodology that described for evolving Rough-Fuzzy Multi layer perceptron with modular concept using a genetic algorithm to obtain a structured network suitable for both classification and rule extraction.

In ٢٠٠٢, Ken McGarry,[٣٣] presented the results of ranking and the analysis of rules extracted from RBF neural networks using both objective and subjective measures. The interestingness of a rule can be assessed by a data driven approach

In ٢٠٠٤, J. Malone, K. McGarry, and C. Bowerman,[٣٢] demonstrated the use of ANFIS to optimize expert's opinions. The ANFIS model offers the advantage of enabling use of initially approximate data in an effective manner whilst, following training, allowing fuzzy rules to be extracted which represent the optimized fuzzy membership functions.

In ٢٠٠٥, J. Malone, K. McGarry, C. Bowerman and S. Wermter,[٣١] have proposed a technique for the automatic extraction of rules from trained SOMs.

١.٣ Thesis objectives

This thesis aims to suggest a new technique which can extract rules from any database from any domain via soft computing methodologies, which involve fuzzy sets, neural networks, genetic algorithms, and their hybridizations. It strives to provide approximate solutions at low cost, thereby speeding up the process. It never depends on the type of data in database or the application domain.

١.٤ Thesis layout

Thesis falls into four chapters; the other three chapters are organized as follow:

Chapter two gives us brief information about KDD process and the significant roles of each of the soft computing methodologies: fuzzy set, neural network and genetic algorithm can be employed in data mining process. It provides us with the existing method of hybridization between these methodologies and the possible role of each one in the data mining fields.

Chapter three (a practical chapter) shows how we can use soft computing methodologies to build an easy model uses that data mining. It explains the structure of the proposed method by making the hybridizing of genetic-neural-fuzzy in high cooperative way.

Chapter four introduces case study of the proposed system in addition to conclusion and future work.

٢.١ Knowledge Discovery in Database (KDD) & Data Mining

KDD refers to the overall process of turning low-level data into high-level knowledge. An important step in the KDD process is ***data mining***. Data mining is an interdisciplinary field with a general goal of predicting outcomes and uncovering relationships in data. It uses automated tools employing sophisticated algorithms to discover hidden patterns, associations, anomalies and/or structure from large amounts of data stored in data warehouses or other information repositories. Data mining tasks can be ***descriptive***, i.e., discovering interesting patterns describing the data, and ***predictive***, i.e., predicting the behavior of the model based on available data [٤٨].

The process of knowledge extraction from databases combines methods of statistical tools, machine learning and databases to find a mathematical and/or logical description, which can be eventually complex, of patterns and regularities in data [٤٠].

The knowledge extraction from a large amount of data should be seen as an interactive and iterative process, and not as a system of automatic analysis. In this way, we cannot simply expect an extraction of useful knowledge by submitting a group of data to a “black box”.

The interactivity of the KDD process refers to the greater understanding, on the part of the users of the process, of the application domain. This understanding involves the selection of a representative data subset, appropriate pattern classes and good approaches to evaluate the knowledge. For a better understanding the functions of the users that use the KDD process, users are divided into three classes: (a) Domain Expert, who should possess a large understanding of the application domain; (b) Analyst, who executes the KDD process and, therefore, he should have a lot of

knowledge of the stages that compose this process and (c) Final User, who does not need to have much knowledge of the domain. Frequently, the Final User uses knowledge extracted from the KDD process to aid him in a decision-making process [٤٨].

The success of the KDD process depends partly on the interaction among users. It is not probable that the Analyst will find useful knowledge in the data without the guarantee of the Expert as to what would be useful for a specific domain. Besides, the interactivity of the process requires that the Final User and the Expert have an effective participation in the choices and decisions during the process [٤٠].

Knowledge discovery from data can be understood as a process that contains, at least, the steps of application domain understanding, selection and preprocessing of data, Data Mining, knowledge evaluation and consolidation and use of the knowledge.

A representative outline containing all these steps is illustrated in Figure (٢-١). The KDD process begins with the understanding of the application domain, considering aspects such as the objectives of the application and the data sources. Next, a representative sample (e.g. using statistical techniques) is removed from database, preprocessed and submitted to the methods and tools of the Data Mining stage with the objective of finding patterns/models (knowledge) in the data. This knowledge is then evaluated as to its quality and/or usefulness, so that it can be used to support a decision-making process. It should be emphasized that, in spite of the visualization tools being used mostly in the knowledge evaluation step, they have great relevance in understanding and evaluating the results of each stage, especially for the Final User[٤٥].

It is important to notice that, because it is an iterative process, the KDD steps are not tight, that is, the correlation among the techniques and methods used in the several stages is considerable, to the point of the occurrence of a small change in one of them affecting substantially the success of the whole process. In this way, the results of a certain stage can change any of the previous stages or even make it necessary restart the whole process [٤٠].

In addition, decision makers are not interested in techniques that rely too much on the underlying assumptions in statistical models. The challenge is not to have any assumptions about the model and try to come up with something new, something that is not obvious or predictable (at least from the decision makers' point of view). Some unobvious thing may have significant values to the decision maker. Identifying a hidden trend in the data or a buried fault in the system is by all accounts a treasure for the investor who knows that avoiding loss results in profit and that knowledge in a complex market is a key criterion for success and continuity. Not with standing, models that are free from assumptions–or at least have minimum assumptions–are expensive to use. The dramatic search space cannot be navigated using traditional search techniques [٤٦].

٢.١.١ The steps in the KDD process

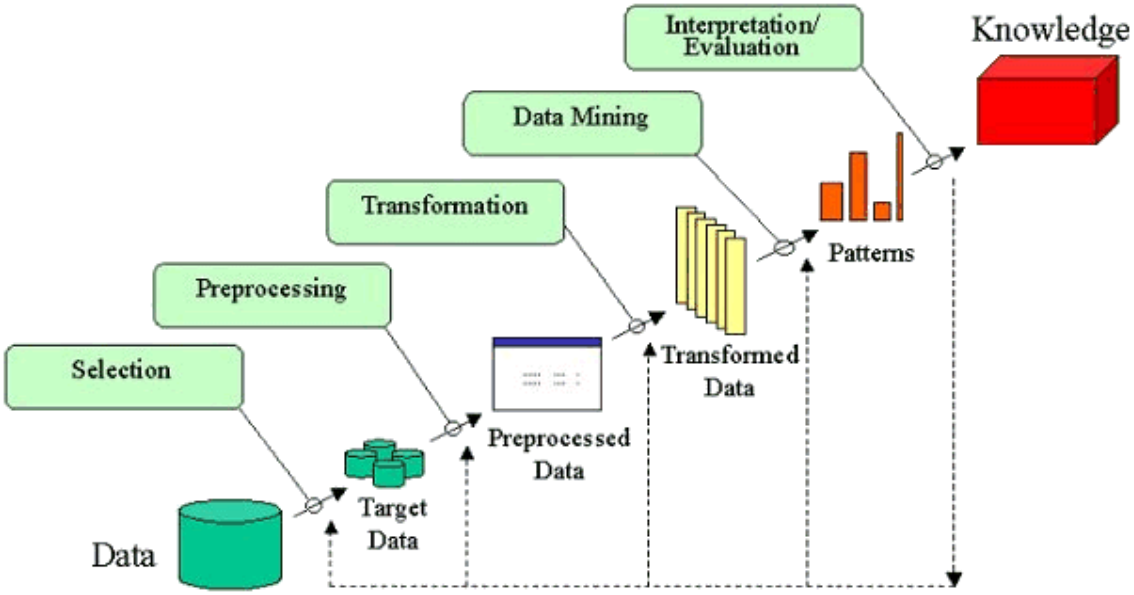

Figure (٢-١) KDD Process main steps [٤٨]

Even though the KDD processes have emerged from different fields, KDD has almost the same steps in all of the different approaches. These steps are [٤٠, ٤٥]:

١. Developing an understanding of the application domain, the relevant prior knowledge, and the goal(s) of the end-user.
٢. Creating a target data set; selecting a data set, or focusing on a subset of variables or data samples, on which discovery is to be performed.
٣. Data cleaning and preprocessing: basic operations such as removing noise or model for noise, deciding on strategies for handling missing data fields, accounting for time sequence information and known changes.
٤. Data reduction: preparing the data set, removing some attributes to suit the set to the goal.
٥. Choosing the data mining task: deciding whether the goal for the KDD process is classification, regression, clustering, etc.

٦. Choosing the data mining algorithm(s): selection method(s) to be used for searching for patterns in the data. This also includes deciding which models and parameters may be appropriate.
٧. Data mining: search for patterns of interest in a particular representational form or set of forms: classification rules or trees, regression or clustering.
٨. Interpreting mined patterns, possible return to any of the steps ١-٧ for further iteration.
٩. Consolidating discovered knowledge.

There is a distinction between information and knowledge. Information is a collection of data, whereas knowledge is some higher understanding that can tell us something more about relations.

A particular data mining algorithm is usually an instantiation of the model/preference/search components. The more common model functions in current data mining practice include the following [٤٣]:

١) Classification: classifies a data item into one of several predefined categorical classes.

٢) Regression: maps a data item to a real valued prediction variable.

٣) Clustering: maps a data item into one of several clusters, where clusters are natural groupings of data items based on similarity metrics or probability density models.

٤) Rule generation: extracts classification rules from the data.

٥) Discovering association rules: describes association relationship among different attributes.

٦) Summarization: provides a compact description for a subset of data.

٧) Dependency modeling: describes significant dependencies among variables.

٨) Sequence analysis: models sequential patterns, like time-series analysis. The goal is to model the states of the process generating the sequence or to extract and report deviation and trends over time.

The rapid growth of interest in data mining is due to the:

١) Falling cost of large storage devices and increasing ease of collecting data over networks;

٢) Development of robust and efficient machine learning algorithms to process this data; and

٣) Falling cost of computational power, enabling use of computationally intensive methods for data analysis.

The notion of scalability relates to the efficient processing of such large data sets while generating from them the best possible models. The most commonly cited reason for scaling up is that increasing the size of the training set often increases the accuracy of learned classification models. In many cases, the degradation in accuracy when learning from smaller samples stems from overfitting, presence of noise, and existence of large number of features. Additionally, scaling up to very large data sets implies that fast learning algorithms must be developed. However, rather than speeding up a slow algorithm, the issue is more of turning an impracticable algorithm into a feasible one[٣٥,٤٥].

The first generation of data mining algorithms has been demonstrated to be of significant value across a variety of real world applications. But these applications work best for problems involving a large set of data collected into a single database, where the data is described by numeric or symbolic features. Here the data invariably does not contain text and image features interleaved with these features, and is carefully and cleanly collected with a particular decision-making task in mind.

Development of new generation algorithms is expected to encompass more diverse sources and types of data that will support mixed-initiative data mining, where human experts collaborate with the computer to form hypotheses and test them. The main challenges to the data mining procedure involve the following[٤٥]:

١) ***Massive data sets and high dimensionality***. Huge data sets create combinatorially explosive search space for model induction, and increase the chances that a data mining algorithm will find spurious patterns that are not generally valid. Possible solutions include robust and efficient algorithms, sampling approximation methods and parallel processing.

٢) ***User interaction and prior knowledge***. Data mining is inherently an interactive and iterative process. Users may interact at various stages, and domain knowledge may be used either in the form of a high-level specification of the model, or at a more detailed level. Visualization of the extracted model is also desirable.

٣) ***Overfitting and assessing the statistical significance***. Data sets used for mining are usually huge and available from distributed sources. As a result, often the presence of spurious data points leads to overfitting of the models. Regularization and resampling methodologies need to be emphasized for model design.

٤) ***Understandability of patterns***. It is necessary to make the discoveries more understandable to humans. Possible solutions include rule structuring, natural language representation, and the visualization of data and knowledge.

٥) ***Nonstandard and incomplete data***. The data can be missing and/or noisy.

٦) ***Mixed media data***. Learning from data that is represented by a combination of various media, like (say) numeric, symbolic, images and text.

٧) ***Management of changing data and knowledge***. Rapidly changing data, in a database that is modified/deleted/augmented, may make previously discovered patterns invalid. Possible solutions include incremental methods for updating the patterns.

٨) ***Integration***. Data mining tools are often only a part of the entire decision making system. It is desirable that they integrate smoothly, both with the database and the final decision making procedure.

٢.٢ Soft Computing For Data Mining

Recently, various soft computing methodologies have been applied to handle the different challenges posed by data mining. The main constituents of soft computing include fuzzy logic, neural networks, and genetic algorithms. Each of them contributes a distinct methodology to addressing problems in its domain. This is done in a cooperative, rather than a competitive, manner. The result is a more intelligent and robust system providing a human-interpretable, low cost, approximate solution, as compared to traditional techniques[٤٥].

Let us first describe the roles and significance of the individual soft computing tools and their hybridizations. It may be mentioned that there is no universally best data mining method; choosing particular soft computing tool(s) or some combination with traditional methods is entirely dependent on the particular application and requires human interaction to decide on the suitability of an approach.

٢.٢.١ *Fuzzy Sets*

The modeling of imprecise and qualitative knowledge, as well as the transmission and handling of uncertainty at various stages are possible through the use of fuzzy sets. Fuzzy logic is capable of supporting, to a reasonable extent, human type reasoning in *natural* form. It is the earliest and most widely reported constituent of soft computing. The development of fuzzy logic has led to the emergence of soft computing [١,٢٩].

Knowledge discovery in databases is mainly concerned with identifying interesting patterns and describing them in a concise and meaningful manner. Fuzzy models can represent a prudent and user-oriented sifting of data, qualitative observations and calibration of common sense rules in an attempt

to establish meaningful and useful relationships between system variables. Despite a growing versatility of knowledge discovery systems, there is an important component of human interaction that is inherent to any process of knowledge representation, manipulation, and processing. Fuzzy sets are inherently inclined toward coping with linguistic domain knowledge and producing more interpretable solutions [٣٢,٣٥].

There is a growing indisputable role of fuzzy set technology in the realm of data mining. Analysis of real-world data in data mining often necessitates simultaneous dealing with different types of variables, *viz.*, categorical/symbolic data and numerical data.

٢.٢.٢ *Neural Networks*

Neural networks were earlier thought to be unsuitable for data mining because of their inherent *black-box* nature. No information was available from them in symbolic form suitable for verification or interpretation by humans. Recently, there has been widespread activity aimed at redressing this situation by extracting the embedded knowledge in trained networks in the form of symbolic rules. This serves to identify the attributes that, either individually or in a combination, are the most significant determinants of the decision or classification. Unlike fuzzy sets, the main contribution of neural nets toward data mining stems from rule extraction and clustering [٢٩].

- ***Rule Extraction****:* [٤١] In general, the primary input to a connectionist rule extraction algorithm is a representation of the trained neural network, in terms of its nodes, links and sometimes the data set. One or more hidden and output units are used to automatically derive the rules, which may later be combined and simplified to arrive at a more comprehensible rule set.

These rules can also provide new insights into the application domain. The use of neural nets helps in ١) incorporating parallelism and ٢) tackling

optimization problems in the data domain. The models are usually suitable in ***data-rich*** environments.

Typically a network is first trained to achieve the required accuracy rate. Redundant connections of the network are then removed using a pruning algorithm. The link weights and activation values of the hidden units in the network are analyzed, and classification rules are generated[٣٩].

٢.٢.٣ *Genetic Algorithms*

GAs are adaptive, robust, efficient, and global search methods, suitable in situations where the search space is large. They optimize a ***fitness function***, corresponding to the preference criterion of data mining, to arrive at an optimal solution using certain genetic operators. Knowledge discovery systems have been developed using genetic programming concepts. The problem addressed is to find common characteristics of a set of objects in an object-oriented database. Genetic programming is used to automatically generate, evaluate, and select object-oriented queries. GAs are also used for several other purposes like fusion of multiple data types in ***multimedia*** databases, and automated program generation for mining multimedia data[١-٥].

٢.٣ Soft Computing –The Hybridization

Soft computing (SC)[١] is an association of computing methodologies centering on ***fuzzy logic*** (FL), ***neural computing*** (NC), ***genetic computing*** (GC) and ***probabilistic computing*** (PC). Collectively, these methodologies provide a foundation for the conception, design and deployment of intelligent systems. The basic idea underlying soft computing is that its constituent methodologies are, for the most part, complementary rather than competitive. The complementarity of the constituents of soft computing implies that their effectiveness may be enhanced by using them in combination rather than isolation. At this juncture, the most visible systems of this combined type are neuro-fuzzy systems. Less visible, but potentially of equal importance are fuzzy-genetic systems. Each of the constituents of soft computing has a set of capabilities to offer. In the case of fuzzy logic, it is the machinery for dealing with imprecision, information granulation and computing with words. For this purpose, the principal tools are provided by fuzzy logic center on the use of linguistic variables and the calculus of fuzzy if-then rules. In the case of genetic computing the principal tool is systematized random search. Below the most known methods of hybridization of these tools:

٢.٣.١ Neural-Fuzzy Computing:

Neuro-fuzzy computation [٢٠, ٢٤] comprises a judicious integration of the merits of neural and fuzzy approaches, enabling one to build more intelligent decision-making systems. This incorporates the generic advantages of artificial neural networks like massive parallelism, robustness, and learning in ***data-rich*** environments into the system. The modeling of imprecise and qualitative knowledge in natural/linguistic terms as well as the transmission of uncertainty is possible through the use of fuzzy logic. Beside these generic

advantages, the neuro-fuzzy approach also provides the corresponding application specific merits as highlighted earlier.

The ***rule generation*** aspect of neural networks is utilized to extract more ***natural*** rules from fuzzy neural networks. The fuzzy MLP [٤٤] and fuzzy Kohonen network [٣١] have been used for linguistic rule generation and inferencing. Here the input, besides being in quantitative, linguistic, or set forms, or a combination of these, can also be incomplete. The components of the input vector consist of membership values to the overlapping partitions of linguistic properties ***low***, ***medium***, and ***high*** corresponding to each input feature. Output decision is provided in terms of class membership values.

٢.٣.٢ *Fuzzy-Genetic Computing:*

Fuzzy-genetic hybridization [١٨,٤٢,٤٥,٤٦,٤٩] is the Systems in which techniques drawn from fuzzy logic and genetic algorithms are used symbiotically to achieve higher levels of performance. Viewed in a more general setting, fuzzy-genetic systems fall within the province of soft computing. There are some aspects of complementarity of FL and GC that are in need of comment. Generally, FL is highly effective in those situations in which there exists a human solution which can be articulated in the language of fuzzy if-then rules. In this sense, FL is for the most part descriptive rather than prescriptive. Relatedly, it is important to note that a human solution may exist even when an objective function or a fitness function cannot be defined precisely. A case in point is the problem of summarization or the much less complex problem of machine translation. The effectiveness of GC depends in large measure on the ability to define a fitness function and simulate system behavior. In many real-world problems this is hard to do. In such cases, a supporting role of GC would be that of local tuning rather than global optimization of the performance of a fuzzy system.

A note worthy point is that any theory and any method can be generalized through fuzzification and/or fuzzy granulation. Fuzzification involves replacement of sets by fuzzy sets while fuzzy granulation implies the use of linguistic variables and fuzzy if-then rules as function approximators. In many cases, fuzzification and fuzzy granulation result in a substantial increase in power and better rapport with reality.

٢.٣.٣ Genetic-Neural Computing

The challenge in building a practical neural network is to choose the right architecture and the right learning parameters. MLP with one hidden layer, using the sigmoid transfer function, could perform any mapping from a set of inputs to the desired outputs. Unfortunately, this tells us nothing about the learning parameters, the necessary number of neurons, or whether additional layers would be beneficial. It is, however, possible to use a genetic algorithm to optimize the network design. A suitable cost function might combine the RMS error with duration of training [٤٥].

Supervised training of a neural network involves adjusting its weights until the output patterns obtained for a range of input patterns are as close as possible to the desired patterns. The different network topologies use different training algorithms for achieving this weight adjustment, typically through back-propagation or errors. However, it is also possible to use a genetic algorithm to train the network. This can be achieved by letting each gene represent a network weight so that a complete set of network weights is mapped onto an individual chromosome. Each chromosome can be evaluated by testing a neural network with the corresponding weights against a series of test patterns. A fitness value can be assigned according to the error so that the weights represented by the fittest generated individual correspond to a trained neural network[٣٠,٤١,٤٤].

٢.٣.٤ Nero-Genetic-Fuzzy Computing:

Now one can extend the power of hybridization of soft computing methodologies from two agents to three or more. a proposed method of hybridizing only three agents and determine its application domain to the data mining specifically to extracting rules from establishments databases.

To study a database all similar records need to group together and then classify them according to preferences attributes. After that, these groups can be described through their preference attributes using IF….THEN rules. Some of the attributes need to be expressed in linguistic terms to gain high meaning description rules.

Artificial Neural Network (ANN)-Kohonen Winner-Take–All was used to cluster (group) the database records. But before doing the clustering, it is important to determine the initial number of clusters and if possible the good initialization of the neural network weights. The Genetic Algorithm (GA) will do this task. When the number of cluster has been determined, the neural network using Winner-Take-All learning rule then will be trained. After training, the network (recall mode) was used to classify the database records. Statistical analysis used to determine the active attributes of each class. Some attributes in linguistic terms can be expressed through using Fuzzy Logic (FL). Suitable rules that describe classes can be. This will be shown in chapter three.

٣.١ Introduction

Databases have grown exponentially in large stores and companies. In the past, system analysts faced many difficulties in finding enough data to feed into their models. Recently, the picture has changed and now the reverse picture is a daily problem–how to understand the large amount of data we have accumulated over the years. Simultaneously, investors have realized that data is a hidden treasure in their companies. With data, one can analyze the behavior of competitors, understand the system better, and diagnose the faults in strategies and systems. Research into statistics, machine learning, and data analysis has been resurrected. Unfortunately, with the amount of data and the complexity of the underlying models, traditional approaches in statistics, machine learning, and traditional data analysis fail to cope with this level of complexity. The need therefore arises for better approaches that are able to handle complex models in a reasonable amount of time.

Human analysts with no special tools can no longer make sense of enormous volumes of data that require processing in order to make informed business decisions. Data mining automates the process of finding relationships and patterns in raw data and delivers results that can be either utilized in an automated decision support system or assessed by a human analyst.

Modern computer data mining systems themselves learn from the previous history of the investigated system, formulating and testing hypotheses about the rules which this system obeys. When concise and valuable knowledge about the system of interest have been discovered, it

can and should be incorporated into some decision support system which helps the manager to make wise and informed business decisions.

The main difference among the various KDD approaches lies thus in the step(s) called ***Data Mining,*** Fig (٣-١); how is it possible to detect patterns in the data in the best way? This has resulted in a number of different algorithms and methods. This may be because the application area is very heterogeneous. There are similarities between a medical database containing records of patients and a cooperation management database, but the differences are usually bigger. Thus, an algorithm proven useful for a medical database may show not to be useful in a cooperate database. There is a quest to find the right method for a specific problem. The ultimate goal must thus be to design methods and algorithms that are universal [٤٨].

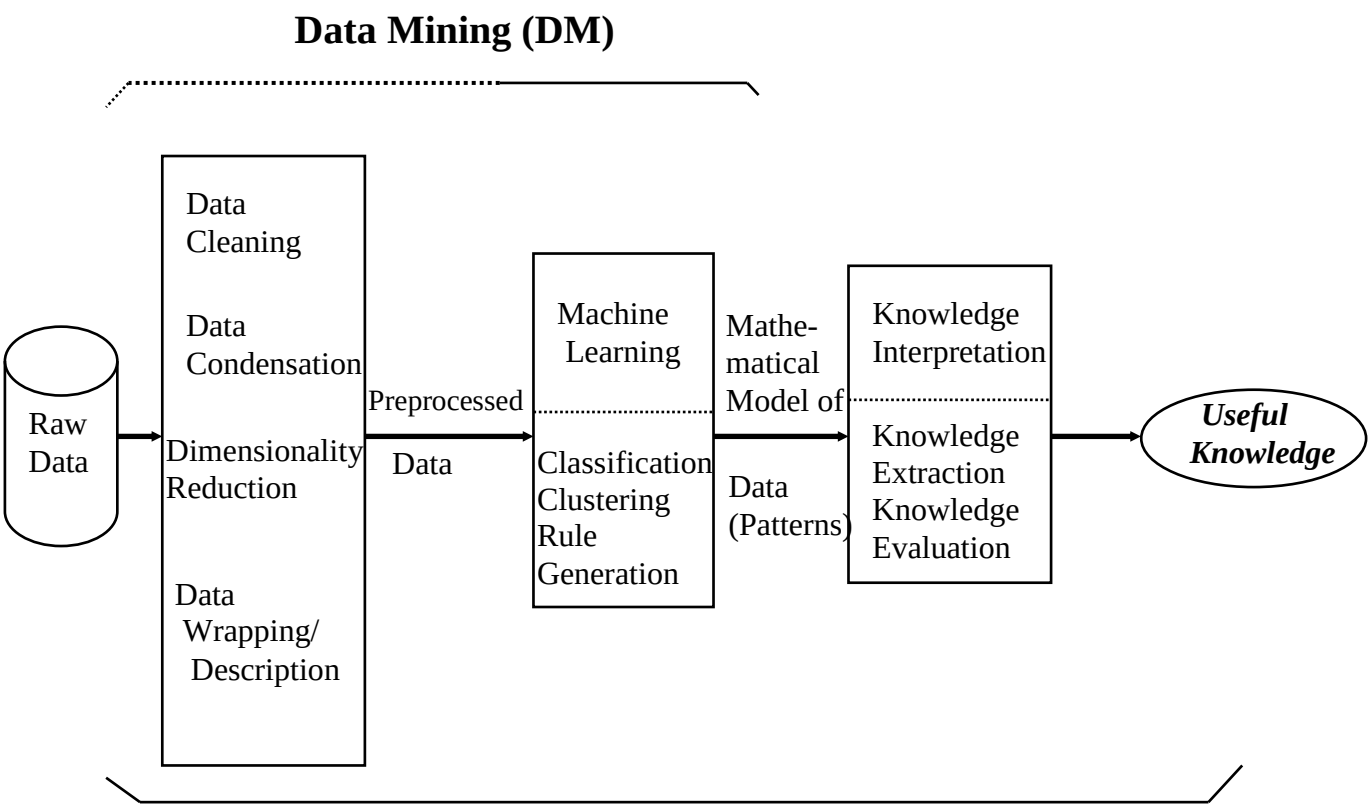

Fig (٣-١) Data Mining & Knowledge Discovery

٣.٢ DBRulesExtractor System.

Soft computing methodologies have been applied to handle the different challenges posed by data mining. The main constituents of soft computing, at this juncture, include *fuzzy logic*, *neural networks* and *genetic algorithms*. Each of them contributes a distinct methodology to addressing problems in its domain. This is done in a cooperative, rather than a competitive, manner. The result is a more intelligent and robust system providing a human-interpretable, low cost, approximate solution, as compared to traditional techniques. Figure (٣-٢) show main procedures of the suggested soft computing approach for solving rules extraction problem.

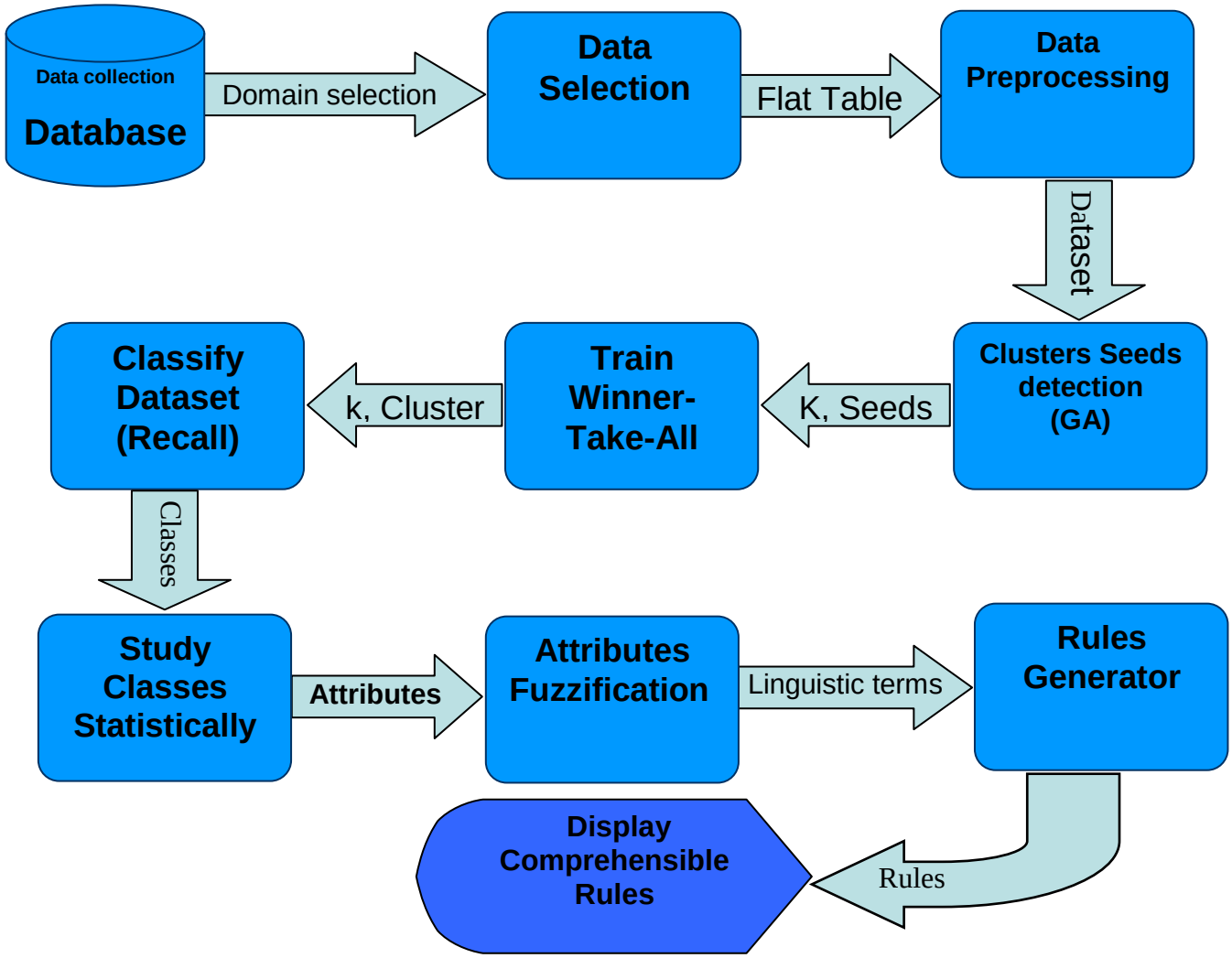

Figure (٣-٢) DBRuleExtractor System diagram

٣.٢.١ Data Collection

Data collection can be a time-consuming and difficult procedure to do correctly, but it is necessary for valid results. If the domain and the analysis to be done are well understood, as in a scientific experiment to test a specific hypothesis, then it is easier to decide what data to collect and how to collect them. But in other cases where the domain is less well understood, where hypotheses may not be clearly specified, then it is difficult to apply the same rigorous data collection methods that are apparent in the former situation. The net effect is that substantial amount of effort has to be devoted to data preparation issues.

Incidental data collection methods refer to the acquisition of data that has originally been collected for one purpose, but is being analyzed for another. A lot of data within organizations are characterized as being of this nature. For example, the responses to surveys may be "reused" for other analysis. Also, the common practice of purchasing data from third party sources is an example of incidental data collection. In practice, many direct marketing campaigns are based on data purchased from third party sources. Relative to active data collection methods, it is more difficult to ensure that data collected via incidental methods are clean. This is because the "history" of such data (i.e., if and how the data have been transformed or aggregated) may not always be known. This can affect analysis - for example linear trends may appear but in actuality is an artifact of a transformation of an underlying process that is non-linear.

Unfortunately, with active data collection methods and especially with incidental data collection methods, it's easy and convenient to assume that data are clean. Assuming this in error can be costly, but

undertaking processes to ensure clean data can also be expensive and time-consuming.

The databases used for testing the system are got from some machine learning web sites that offer such type of data for student. Every database has been attached with enough prior information that helps us in using it. They belong to more than one application domain such as medicine, housing, and cars data.

٣.٢.٢-Selection

Features (attributes) Selection methods in Data Mining and Data Analysis problems aim at selecting a subset of the variables, or features, which describe the data in order to obtain a more essential and compact representation of the available information. The selected subset has to be small in size and must retain the information that is most useful for the specific application. The role of Feature Selection is particularly important when computationally expensive Data mining tools are used, or when the data collection process is difficult or costly. Feature Selection problems are typically solved in the literature using search techniques, where the evaluation of a specific subset is accomplished by a proper function (filter methods) or directly by the performance of a Data Mining tool (wrapper methods). Feature selection has been an active research area in pattern recognition, statistics, and data mining communities.

For simplicity reason Data has been selected manually using a suitable interface that allow user to brows storage device to choose which database and which table or query to use. Also using the same interface fields can be selected or deselected depending on the effectiveness of each fields in expected clusters. The choosing of interesting fields is

depending on one or more of many reasons. Like as prior information about domain, by asking the interesting people of domain, through display the result of several running of the system on the same data, or by inspecting the data manually. See appendix A.

٣.٢.٣-Preprocessing

Preprocess data needed since data quality is a key issue with data mining and to increase the accuracy of the mining has to perform data preprocessing. The researchers in data mining fields find that ٨٠٪ of mining efforts are often spent on data quality. So, how it is possible to preprocess data? This is done through: Data Cleaning, Data Integration, Data Reduction, and Data Normalization.

٣.٢.٣.١ Data Cleaning: Always real-world data are:

- Incomplete: – missing values, missing attributes, or containing only aggregate data. To handle Missing Values:
  ١. Use attribute mean for all samples belonging to same class.
  ٢. Use most probable value based on existing data
     ex.: What would probably be the salary of a person with age *x* and education *y* based on the other data we currently have?
- Noisy: – Data may containing errors or outliers within it:

  To detect noisy data use:
  a. Histogram - data distribution analysis
  b. Cluster Analysis- by detecting data that are outside any cluster. Find clusters and look for elements outside of any cluster.

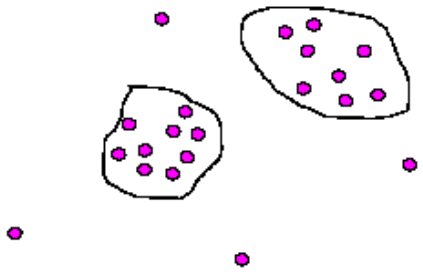

Figure (٣- ٣) Outliers points

c. Regression- by using regression function.

a. Find “best fitting” curve to existing data points.

b. Points not matching curve are outliers.

Example: y=x is best fitting curve for current data. The outliers are the three points outside of the curve.

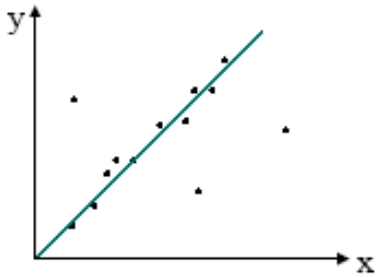

Figure (٣-٤) Outliers Points

To Smooth Noisy Data:

a. Binning- by arranging the data into buckets will reduces distinct values and gets rid of outliers:

Step ١: Partition sorted values into equal size bins.

Step ٢: Smooth by bin means/medians/boundaries.

Example:

- ٤,٨,١٥,٢١,٢١,٢٤,٢٥,٢٨,٣٤

  Bin ١: ٤, ٨, ١٥

  Bin ٢: ٢١, ٢١, ٢٤

  Bin ٣: ٢٥, ٢٨, ٣٤

- By bin mean:

  Bin١: ٩, ٩, ٩; bin ٢: ٢٢, ٢٢, ٢٢ ; bin ٣: ٢٩, ٢٩, ٢٩

- Smoothing by bin boundary

  Bin ١: ٤, ٤, ١٥; bin ٢: ٢١, ٢١, ٢٤; bin ٣: ٢٥, ٢٥, ٣٤

b. Concept Hierarchy.

Example: presenting numeric values such as age as young, middle age, and old.

c. Ignoring outliers detected by Histogram, Clustering, Regression(Outliers are data that are outside the range of or inconsistent with the remaining data)

- Inconsistent: – containing discrepancies in codes or names.

To Handle Inconsistent Data:

a. Use known Functional dependencies example: item# → item
b. Revisit data integration, as some inconsistencies might exist because of different names of the same attribute.

٣.٢.٣.٢ Data Integration

Consolidate different source into one repository, usually data warehouse (schema re-consolidation)

١. Using metadata
٢. Correlation analysis (measure how strongly one attribute implies the other attribute).

٣.٢.٣.٣ Data Reduction

To increase the efficiency we can reduce the huge data set to smaller representative methods:

١. Data aggregation (data cubes) example: number of items sold in year vs. in month.
٢. Dimension/attribute reduction
٣. Data Compression
٤. Discretization:
   a. Discretization is to transform the numeric (Continuous) data to Categorical values.
   b. Some data Mining Algorithms only accept categorical values.
      Example: Continues data: ١,٢,٣,٤,٥,…,٢٠
      Discretized values: ١-٥; ٦-١٠; ١١-١٥; ١٦-٢٠
      Continuous data for feature Age: ١,…,٩٩
      Categorical values: ١-١٥: assign this range to concept "child"
      ١٦- ٤٠: assign this range to concept "Young" and so on

٣.٢.٣.٤ Data Normalization

١. Scale the data value to a range using methods such as:
   a. Min-Max :
      A. Linear transformation of the original input range into a newly specified data range (typically ٠-١).

$$y = ((y-min)/(max-min))*(max' - min') + min' \quad \text{..... (٣-١)}$$

Where: ***min*** is old minimum value, ***min'*** is new minimum, ***max*** is old maximum, ***max'*** is new maximum.

Consider old data that ranged from ٠-١٠٠, we now obtain an equation to migrate it to ٥-١٠ range.

example:

y' = (y/٢٠) + ٥

y = ٠, y'= ٥

y= ١٠, y'= ٥.٥

y = ٩٠, y' = ٩.٥

b. Z-Score

A. Useful when min and max are unknown or outliers dominate the value min-max.

B. The goal is that most of the data will lie within the origin to a standard deviation.

C. If the majority of data falls within ٥٠ and ١٠٠, but you have a few data points outside that range, z score will compress most of the data into a small range.

y'=(y-mean)/std ……..……(٣-٢)

Where ***mean*** is a mean of that variable and ***std*** is a standard deviation of it

c. Decimal Scaling

A. Divide the value by ١٠^{n} where n is the number of digits of the maximum absolute value.

Y'=(y/١٠^{n})……...(٣-٣)

Example: X=٩٠٠ is maximum value

n = ٣

٩٠٠ scales to ٠.٠٠٩.

٣.٢.٣.٥ Linguistic Terms to Numeric Format (Coding):

Linguistic variables have to convert to continuous values to make a suitable format when treated with machine learning algorithms. Then we use appropriate method to transform these variables to a continuous form.

To code attributes of linguistic variable one can use following procedure:

(١) Create the repetition table by determining the repetition times for each linguistic term.

(٢) Rearrange the table by making the large value repetition in the middle and the lesser on right and left of it until minimum repetition becomes at most left and most right.

(٣) Assign code for each linguistic term depending on its new order in the repetition table. Figure (٣-٥)

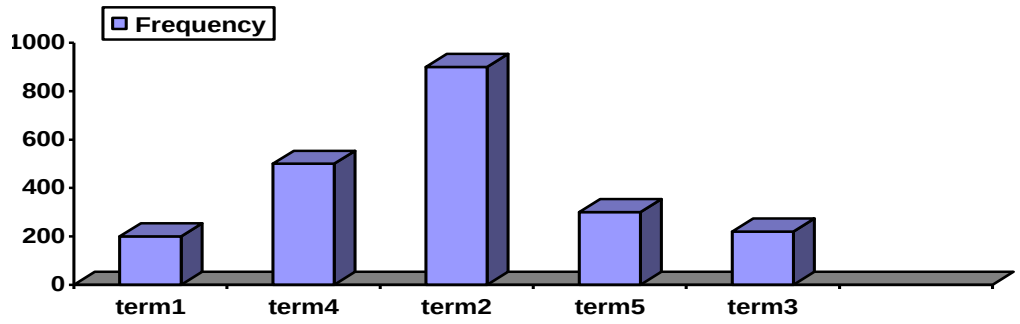

Figure (٣-٥) Terms Repetition Graph

٣.٢.٤-Data Mining – Patterns Discovery

Data mining consists of the (semi-)automatic extraction of knowledge from data. This statement raises the question of what kind of knowledge we should try to discover. Although this is a subjective issue, we can mention three general properties that the discovered knowledge should satisfy; namely, it should be accurate, comprehensible, and interesting.

In data mining we are often interested in discovering knowledge which has a certain predictive power. The basic idea is to predict the value that some attribute(s) will take on in "the future", based on previously observed data. In this context, we want the discovered knowledge to have a high predictive accuracy rate.

The discovered knowledge wanted to be comprehensible for the user. This is necessary whenever the discovered knowledge is to be used for supporting a decision to be made by a human being. If the discovered "knowledge" is just a black box, which makes predictions without explaining them, the user may not trust it [٤٨]. Knowledge comprehensibility can be achieved by using high-level knowledge representations. A popular one, in the context of data mining, is a set of IF-THEN (prediction) rules, where each rule is of the form:

IF <some_conditions_are_satisfied>
THEN <its_belong_to_cetain_class >

The third property, knowledge being interesting, is the most difficult one to define and quantify, since it is, to a large extent, subjective.

٣.٢.٤.١ Clusters Seeds Detection

Clustering [٧,٩] is a popular unsupervised pattern classification technique which partitions the input space into *K* regions based on some similarity/dissimilarity metric. The number of partitions/clusters may or may not be known a priori. Let the input space *S* be rerepresented by *n* points *{x١, x٢....., xn}*, and the *K* clusters be represented by $C_١$, $C_٢$, ..., C_K . Then

$C_i \neq ٠$ for $i=١,\ldots,K$

$C_i \cap C_j = \varnothing$ for $i=١,\ldots..,K$, $j=١,\ldots..,K$, $i \neq j$, and

$$\bigcup_{i=1}^{k} C_i = S.$$

A ***kohonen winner-take-all network*** used to cluster a database records. This network classifies input vectors into one of the specified number of ***K*** categories according to the clusters detected (genetic algorithm has been used to determine *K*) in the training set {x١,x٢,…,xN}. The training is performed in an unsupervised mode, and the network undergoes the self-organization process[٢٤]. During the training, dissimilar vectors are rejected, and only one, the most similar, is accepted for weight building. As mentioned before it is impossible in this training method to assign network nodes to specific input classes in advance. It is equally impossible to predict which neurons will be activated by members of particular cluster at the beginning of the training. This node to cluster assignment is easily done by calibrating the network after training.

In most real life situations the number of clusters in a data set is not known a priori. The real challenge in this situation is to be able to automatically evolve a proper value of *K* as well as providing the

appropriate clustering. So, before we train this network we need to determine the number of clusters *K* and the optimal weight can be initialzed to the network. Genetic algorithms (GA) based clustering technique can automatically evolve the appropriate clusters number of a data set. The chromosome encodes the centres(weight) of clusters(nodes), whose value may vary. Modified versions of crossover and mutation operators are used.

Genetic Algorithms (GAs) belong to a class of search techniques that mimic the principles of natural selection to develop solutions of large optimization problems. GAs operate by maintaining and manipulating a population of potential solutions called chromosomes. Each chromosome has an associated fitness value which is a qualitative measure of the goodness of the solution encoded in it. This fitness value is used to guide the stochastic selection of chromosomes which are then used to generate new candidate solutions through crossover and mutation. Crossover generates new chromosomes by combining sections of two or more selected parents. Mutation acts by randomly selecting genes which are then altered; thereby preventing suboptimal solutions from persisting and increases diversity in the population. The process of selection, crossover and mutation continue for a fixed number of generations or until a termination Condition is satisfied. GAs have applications in fields as diverse as VLSI design, pattern recognition, image processing, neural networks, machine learning, etc.

GA algorithm used to find optimal clusters' seeds & their number according to syntax chart of the algorithm is shown in the figure (٣-٦).

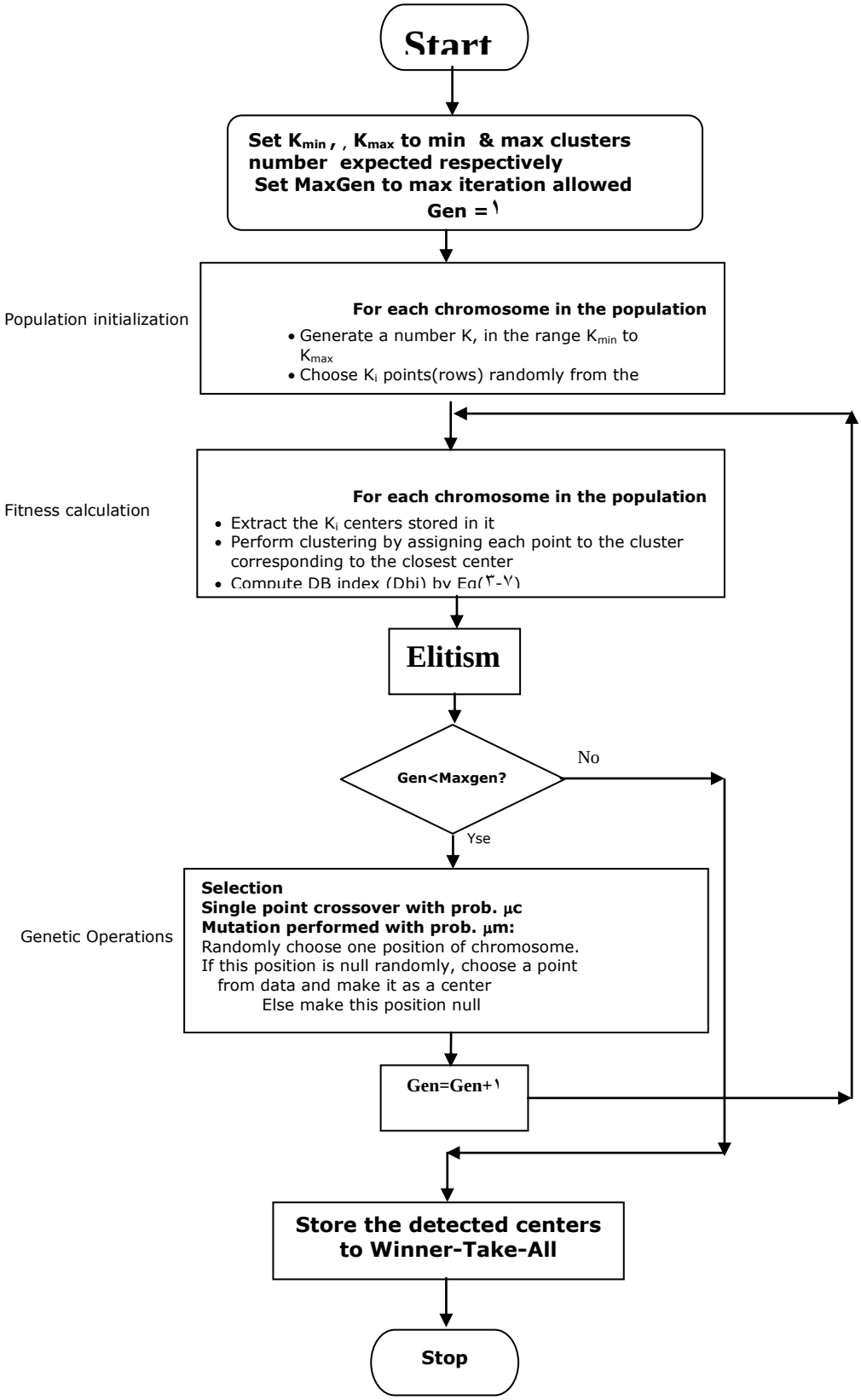

Figure (٣-٦) Flowchart of GA-Based Clusters detection

A.Representation (encoding of solution)

The chromosomes are made up of a list pointers. If the pointer at any gene is not null, that means there is a supposed center. This center is drawn randomly from the data set. On the other hand gene (pointer) with null mean has had no center encoded in it. The value of *K* is assumed to lie in the range [K_{min}; K_{max}], where K_{min} is chosen to be ٢ unless specified otherwise. The length of a string is taken to be K_{max} where each individual gene position represents either apointer to actual center or a null.

B.Population initialization

For each string *i* in the population (*i*= ١,……, *P*, where *P* is the size of the population), a random number K_i in the range [K_{min} - K_{max}] is generated. This string is assumed to encode the centres(each center represent a weights of node of Kohonen winner-take-all network) of K_i clusters. For initializing these centres, K_i points are chosen randomly from the dataset. These points are distributed randomly in the chromosome. Let us consider the following example.

Example: Let K_{min} =٢ and K_{max} =١٠. Let the random number K_i be equal to ٤ for chromosome *i*. Then this chromosome will encode the centres of ٤ clusters. Let the ٤ cluster centres (٤ randomly chosen points from the data set) be (١٠:٠, ٥:٠) (٢٠:٤, ١٣:٢) (١٥:٨, ٢:٩) (٢٢:٧, ١٧:٧). On random distribution of these centres in the chromosome, it may look like :

[null,↑(٢٠:٤;١٣:٢),null,null,↑(١٥:٨;٢:٩) ,null,↑(١٠:٠;:٠)↑(٢٢:٧;١٧:٧),
null,null].

C.Fitness computation

For each cromosome in the population build a Kohonen Winner-Take-all network from the data encoded in that cromosome. Run all data set on this network. The fitness of a chromosome is computed using the

Davies–Bouldin index (DBi). DBI is determined as follows [١٠] : Given a partition of the N points into K clusters, one first defines the following measure of within-to-between cluster spread for two clusters, C_j and C_k for ١ ≤ j, k ≥ K and j ≠ k.

$$R_{jk} = \frac{e_j + e_k}{D_{jk}} \quad \text{.....(٣-٤)}$$

where e_j and e_k are the average dispersion of C_j and C_k, and D_{jk} is the Euclidean distance between C_j and C_k. If m_j and m_k are the centers of C_j and C_k, consisting of N_j, and N_k points respectively:

$$e_j = \frac{1}{N_j} \sum_{x \in C_j} \left\| x - m_j \right\|^2 \quad \text{.....(٣-٥)}$$

and $D_{jk} = \left\| m_j - m_k \right\|^2$

And the term R_k for each C_k id defined as

$$R_k = \max_{j \neq k} R_{j,k} \quad \text{.....(٣-٦)}$$

Then that the DBi is defined as:

$$DB(K) = \frac{1}{k \sum_{k=1}^{K} R_k} \quad \text{.....(٣-٧)}$$

The DBi can be incorporated into any clustering algorithm to evaluate a particular segmentation of data. The *DBi* takes into account cluster dispersion and the distance between cluster means. Well-separated compact clusters are preferred. The DBI favors small numbers of clusters. Optimizing the DBI frequently eliminates clusters by forcing them to be empty.

The objective is to minimize the *DB* index for achieving proper clustering. The fitness function for chromosome *j* is defined as ١/*DBj* ,

where *DBj* is the Davies–Bouldin index computed for this chromosome. The maximization of the fitness function will ensure minimization of the DB index.

D.Genetic Operations

The following genetic operations are performed on the population of strings for a number of generations.

*a.**Selection***: Conventional proportional selection is applied on the population of strings. Here, a string receives a number of copies that is proportional to its fitness in the population.

*b.**Crossover***: During crossover each cluster centre is considered to be an indivisible gene. Single point crossover, applied stochastically with probability ηc, is explained below with an example.

Example: Suppose crossover occurs between the following two strings:

null, ↑ , null, null, ↑ ,| null , ↑ , ↑ ,null, null

null, ↑ , null, ↑ , null,| ↑ , ↑ ,null, ↑ ,null

Let the crossover position be ۰ as shown above. Then the offspring are:

null, ↑ , null, null, ↑ , ↑ , ↑ ,null, ↑ ,null

null, ↑ , null, ↑ , null, null , ↑ , ↑ ,null, null

*c.**Mutation***: Each position in a chromosome is mutated with probability η_m in the following way. If the value at that position is not null , then it becomes null else new cluster center is created by selecting random points from dataset and making the pointer point to it.

***E.Termination criterion**:* we compute the MSE for each cromosome(Kohonen Winner-Take-All network) and the processes of fitness computation, selection, crossover, and mutation are executed until

this measure becomes below some predefined threshold. Also the generations count is used to avoid non stoping process. The best string having the largest fitness (i.e., smallest DB index value) seen up to the last generation provides the solution to the clusters count problem and the seeds refrences.

elitism implemented at each generation by preserving the best string seen up to that generation in a location outside the population. Thus on termination, this location contains the centres of the final clusters of course with their count.

F.Elitism : When creating new population by genetic algorithm processes, we might lose the best chromosome since the selection of chromosomes (or candidate solutions) is more or less done at random. Elitism is the name of method, which first copies the best chromosome (or a few best chromosomes) to new population for further evolution. Elitism can very rapidly increase performance of GA because it prevents losing the best found solution. See Figure (A-٣)

After we get the number of cluster excpected in the dataset and the good inialization weight from seeds refrences, one can train kohonen winner-take-all network.

٣.٢.٤.٢ Training Unsupervised Kohonen Network.

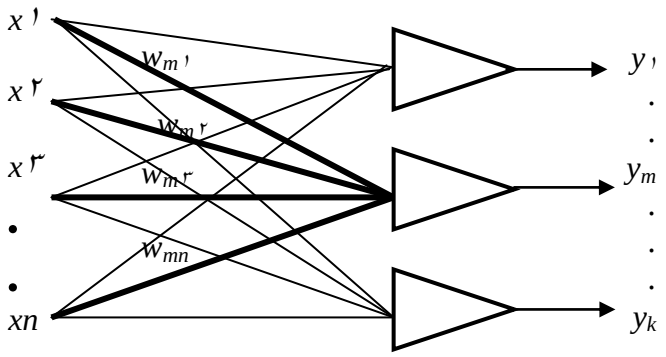

Fig (٣-٧)Kohonen Winner-Take-All learning rule

The processing of input data x from the training set { $x_1, x_2, ..., x_N$} which represnts *K* clusters[٢٦] follows the customray expression

$$y = \Gamma[Wx] \quad \text{.....(٣-٨)}$$

with diagonal elemets of the operator Γ being continuous activation functions operating componentwise on entries of vector Wx. The processing by the layer of neorons is instaneous and feedforward. To analyze network performence, we need to rearrange the matrix W to the following form:

$$W = \begin{bmatrix} w_1^t \\ w_2^t \\ \vdots \\ w_K^t \end{bmatrix} \quad \text{.....(٣-٩)}$$

Where

$$w_i = \begin{bmatrix} w_{i1} \\ w_{i2} \\ \vdots \\ wn \end{bmatrix} \text{ for } i = ١, ٢, ..., K \quad \text{.....(٣-٩a)}$$

is the column vector equal to the i'th row of the weight matrix W. Component weights of w_m are highlighted in figure (٣-٧) showing a winner take all learning network. The learning algorithm treats the set of *K* weight vectors as variable vectors that need to be learned. Prior to the learning, the normalizaton of all weight vectores is required:

$$\hat{w}_i \overset{\Delta}{=} \frac{w_i}{\|w_i\|}, \text{ for } i = ١, ٢, .., K$$

The weight adjustment criterion for this mode of training is the selection of $\hat{w}_i$ such that

$$\left\| x - \hat{w}_m \right\| = \min_{i=1,2,..,K} \left\{ \left\| x - \hat{w}_i \right\| \right\} \quad \text{.....(٣-١٠)}$$

The index *m* denotes the wining neuron number corresponding to the vector w_m,which is the closest approximation of the current input x, let us see how this learning should proceed in terms of weight adjusment. The left side of the (٣-١٠) equation can be rearranged to the form

$$\left\| x - \hat{w}_m \right\| = (x^t x - 2\hat{w}_m^t x - 1)^{1/2} \quad \text{.....(٣-١١)}$$

It is obvious from eq(٣-١١) that searching for the minmum of *K* distance as on the right side of equation above corresponds to finding the maximum among the *K* scalar products

$$\hat{w}^t{}_m = \max_{i=1,2,...,K} (\hat{w}_i^t x) \quad \text{.....(٣-١٢)}$$

The left side of equation (٣-١٢) is the activation value of the "wining" neron which has the largest value net_i, *i*=١,٢,….,*K*. When using the scalar product metric of simiarity as in the equation the synaptic weight vectors should be modified accordingly so that they become more similar to the current input vector. With the similarity criterion being $\cos\psi$, the weight vector lengths should be identical for this training approach. However, their directions should be modified. Intutivly, it is clear that a very long weight vector could lead to a large output of its neuron even if there were a large angle between the weight vector and the pattren. This explains the need for weight normalization.

After the wining neuron has been identified and decleared a winner, its weights must be adjusted so that the distance is reduced in the current training step. Thus, $\|x - w_m\|$ must be reduced, preferably along the gradient direction in the weight space $w_{m١}, w_{m٢}, \ldots, w_{mn}$

$$\nabla_{wm} = \|x - w_m\|^2 = -2(x - w_m) \quad \text{.....(٣-١٣a)}$$

Scince vectors x have a certain probability distribution within a cluster and we are dealing with a single adjusment step due to the single realization of the input, only a fraction of the increment in equation should be involved in producing the sensible weight adjustments. It seems reasonable to reward the weights of the winning neuron with an increment of weight in the nagative direction, thus in the direction $x\text{-}w_m$, we have:

$$\Delta \hat{w}_m = \alpha(x - \hat{w}_m), \quad \text{(٣-١٣b)}$$

where α is a small learning constant selected heuristically, usualy between ٠.١ and ٠.٧. the remaining weight vectors $\hat{w}_i, i \neq m$, are left unaffected.

Using a superscript to index the weight updates and restating the update criterion (٢), the learning rule (٥b) in the *t*'th step can be rewritten in a more formal way as follows

$$\hat{w}_m^{t+1} = \hat{w}_m^{t} + \alpha^t (x - \hat{w}_m^{t}) \quad \text{(٣-١٣c)}$$

$$\hat{w}_i^{t} = \hat{w}_i^{t}, \qquad for\ i \neq m \quad \text{(٣-١٣d)}$$

where α^t is a suitable learning constant and m is the number of winning neuron selected based on scalar product comparison as in Eq(٣-١٠).While learning continues and clusters are developed, the network weights acquire similarity to input data within clusters. To prevent further unconstrained growth of weights, α is usually reduced monotonically and the learning slows down.

Learning according to Eq(٣-١٢) and (٣-١٣) is "winner-take-all" learning, and it is a common competitive and unsupervised learning technique. The

winning node with the largest net_i is rewards with a weight adjusment, while the weights of others remain unaffected.

This mode of learning is easy to implement as computer simulation; one merely searchs for the maximum response and rewards the winning weights only, in a real network it is possible to implement a winner-take-all layer by using units with lateral inhibition to the other neurons in the form of inhibitory connections. At the same time, the neuron should possess excitatory connections to itself

Let us look at the impact of the learning rule Eq(٣-١٣c) and Eq(٣-١٣d) on the performance of the network. The rule should increase the chance of winning by the m'th neuron as in Eq(٣-١٢) for reptition of the same input pattren using the updated weights. If the requirement holds, then inequality (٣-١٤a) should be valid for new weights $\hat{w}_m^t$,

$$\hat{w}_m^t x < (\hat{w}_m^t + \Delta \hat{w}_m^t)x \quad \text{.....(٣-١٤a)}$$

Using (٣-١٣b) we obtain

$$\Delta \hat{w}^t_m sx > 0 \quad \text{.....(٣-١٤b)}$$

Or

$$x^t x - w'^t_m > 0 \quad \text{.....(٣-١٤c)}$$

Which is equivalent to

$$\left\|x^t\right\|.\|x\|\cos\theta - \left\|\hat{w}_m^t\right\|.\|x\|\cos\varphi > 0 \quad \text{.....(٣-١٤d)}$$

Assuming normalized vectores $\hat{x}$=x reduces (٣.١٤d) to

$$١\text{-}\cos\psi > ٠ \quad \text{.....(٣-١٤e)}$$

where $\psi = \angle(\hat{w}, \hat{x})$. Since (٣-١٤e) is always true, the winner-take-all learning rule produces an udate of the weight vector in the proper direction.

It is instructive to observe the geometrical interpretation of the rule. Assume that in this step the normalized input vector is denoted as $\hat{x}$ and the vector $\hat{w}_m$ yield the maximum scalar product $\hat{w}_i^t \hat{x}$, i=١,٢,….,*K*. Next, a difference vector $\hat{x} - \hat{w}_m$ is created as shown. To implement the rule of eq(٣-١٣c) and (٣-١٣d) for x=$\hat{x}$, an increment of the weight vector is computed as a fraction of $\hat{x} - \hat{w}_m$. The result of weight adjusment in this training step is mainly the rotation of the weight vector $\hat{w}_m$ toward the input vector without a significant length change. The adjusted weight vector results as w'_m and is of a length below unity. To begin with the new training step, w'_m must be normalized. Let us notice that another input beloning to the m'th cluster would make the vector w_m even more representaive of the cluster m.

In the long term, this learning mode leads to the weight vectors that approximate the ensemble of past winning input vectors. However, scince the weights are adjusted in proportion to the number of events that end up with weight adjusments, this nework reacts to the probability of occurrence of inputs. In this context, the network may be used as a clustering network for the paricular probability of training vectors coming from each cluster. After the learning is completed, each $\hat{w}_i$ represnts the centroid of an i'th decision region, i=١,٢,….,*K*, created in the n dimensional space of pattren data. On the other side, the network possesses an interesting feature sensitivity.

To sum up, vectors $\hat{w}$ after training will become organized much like the set of example vectors $\hat{x}$ used for training.

Initialization of Weights.

Preferably random initial weight vectors would be used for this training. This indicates that initial weights should be uniformly distributed on the unity hypersphere in n-dimensional pattren space. Self organization of the network using the described training suffers from some limitations, however . One obvious deificiency related to a single-layer architecture is that lineaely nonseparable pattrens cannot be efficiently handled by this network. The second limitation is that network training may not always be successful even for linearly separable pattrens. Weights may get stuck in isolated regions without forming adequte clusters. In such cases the training must be reinitialized with new initial weights, or noise superimposed on weight vectors during training. After the weights have been trained to provide coarse clustering, the learning constant α should be reduced to produce finer adjusments. This often results in finer weight tuning within each cluster.

One of the weight selection methods devloped for training the network is called *convex combination.* In this method all weight vectors are initialized at value $w_i^0 = \frac{1}{\sqrt{n}}[1 \quad 1 \quad \cdots \quad 1]^t$ for i=١,٢,…,K ……(٣.١٥)

The training starts at the weights as above and proceeds as in expression (٣-١٣c) and (٣-١٣d) with very low α value. This forces the weight vectors at the beginning of learning to be close to the input vectors and to have equal lengths. As learning progresses, α is slowly increased. This allows for the gradual separation of weights according to the input

clusters used for training. This procedure improves the chances for successful training, but does slow down the process.

Another approach used for weight initialization by using the centers values that result from the process of finding the number of cluster by GA. It's known that the winner-take-all network tries to store the mean value of each attribute beloning to some cluster in the weights of this node. The centers values are copied from GA into weight vectors.

٣.٢.٤.٣ Classify Dataset into Detected Classes.

The network trained in the winner-take-all mode responds instantaneously during feedforward recall at all *K* neuron outputs. The response is computed according to (٣-٩). Note that the layer now performs as a filter of the input vectors such that the largest output neuron is found as follows

$$y_m=\max(y_1,y_2,\ldots,y_K)$$

and the input is identified as beloning to cluster *m*. In general, the neurons activation functions should be continuous in this network. For some applications, however, $y_m=1$ and $y_i=0$, $i\neq m$, must be set in the recall mode of the clustering layer. In this way, for example, the weights of the following layer can be fanned out from the activated node of this previous layer while other nodes remain suppressed.

Before a one-to-one vector-to-cluster mapping can be made after the network is trained in the unsupervised mode, it needs to be calibrated in a supervised enviorment. The calibration involves the teacher applying a sequence of *K* best matching class/cluster inputs and labeling the output nodes ١,٢,….,*K* ,respectively, according to their observed responses. Obviously, the calibrating labels assigned to the physical neruons of the

layer would vary from training to training depending on the sequence of data within the training set, the training parameters, and the initial weights. Once the clustering network is labled, it can perform as a cluster classifier in a local representation.

٣.٢.٤.٤ Classes' Statistical Attributes

After classification has been made through Kohonen winner-take-all network, one can inspect each class of the pattrens acquired. There are several methods of inspections. The easiest one is statistical anlysis of each class. Using central tendency and dispertion statistical measures one can form several rules that govern each class attributes.

a. Measures of Central Tendency: Measures of central tendency are measures which are representative of a sample or population. They enable one to be more objective when drawing conclusions or making inferences. These measures identify the *center* or *middle* of a set of values and best characterize the distribution. The typical measures of central tendency are:

A. Mode: Value which occurs most often. It is the most typical category

B. Median: Value corresponding to the middle case or middle observation

C. Mean: Arithmetic average

Mean= sum_of_all_values / number_of_all_values

b. Measures of Dispersion: Another important characteristic of a data set is how it is distributed, or how far each element is from some measure of central tendency (average). There are several ways to measure the variability of the data. Although the most common and

most important if which is the standard deviation, which provides an average distance for each element from the mean, several others are also important, and are hence discussed here.

١. Range: is the difference between the highest and lowest data element. Symbolically, range is computed as x_{max}-x_{min}, although this is very similar to the formula for midrange. This is not a reliable measure of dispersion, since it uses only two values from the data set. Thus, extreme values can distort the range to be very large while most of the elements may actually be very close to each other. For example, the range for the data set ١, ١, ٢, ٤, ٧ introduced earlier would be ٧-١=٦

٢. Standard Deviation: The Standard deviation is another way to calculate dispersion. This is the most common and useful measure because it is the average distance of each score from the mean. The formula for standard deviation is as follows.

$$\sigma = \sqrt{\frac{\Sigma(\mu - x_i)^2}{N}} \quad \text{.....(٣-١٥)}$$

٣. Variance: Variance is the third method of measuring dispersion. In fact variance is just the square of the standard deviation

$$\sigma^2 = \frac{\Sigma(\mu - x_i)^2}{N} \quad \text{.....(٣-١٦)}$$

٣.٢.٤.٥ Attributes Fuzzification

After the attribute of each class has been determined, these attribute always take a numeric format. Actually, to make the rules more understandable it should wrote in linguistic terms instead of numeric values. This done through fuzzification.

The purpose of the fuzzification process is to allow a fuzzy condition in a rule to be interpreted. For example the condition 'person = tall' in a rule can be true for all values of 'height'. A person who is ١٨٠ cm in height is 'tall' with a confidence factor of ٠.٥ (membership value of the club 'tall'). It is the gradual change of the membership value of the condition 'tall' with height that gives fuzzy logic its strength.

Normally fuzzy concepts have a number of values to describe the various ranges of values of the objective term which they describe. For example, the fuzzy concept 'tallness' may have the values 'Tall', 'Medium height' and 'Short'. Typically, the membership functions of these values are as shown in the graph below:

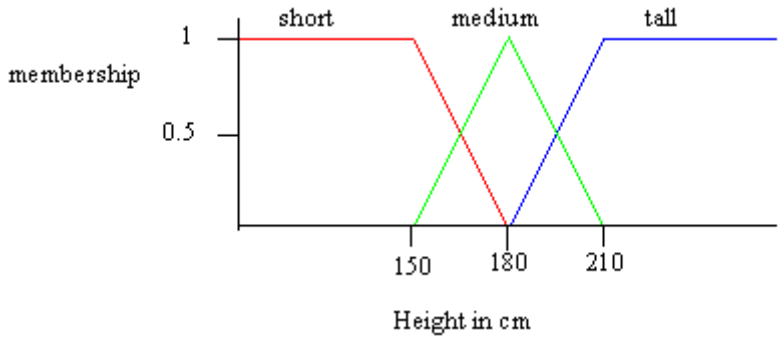

Figure (٣-٨) Fuzzy membership function for "tallness"

Typically, fuzzy concepts have an odd number of values; ٣, ٥ or ٧. One can extend the above values by adding very short and very tall.

The real power of fuzzy logic systems, compared to crisp logic systems, lies in the ability to represent a concept using a small number

of fuzzy values. This therefore reduces the number of rules required to capture the knowledge relating to that concept.

٣.٢.٥ Rules Induction

A data mine system has to infer a model from the database, that is, it may define classes such that the database contains one or more attributes that denote the class of a tuple, i.e., the predicted attributes, while the remaining attributes are the predicting attributes. Class can then be defined by condition on the attributes. When the classes are defined, the system should be able to infer the rules that govern classification. In other words the system would find the description of each class.

Production rules have been widely used to represent knowledge in expert systems and they have the advantage of being easily interpreted by human experts because of their modularity, i.e., a single rule can be understood in isolation and does not need reference to other rules. The prepositional like structure of such rules has been described earlier but can summed up as if-then rules.

After calculating these statistical measures for each class and fuzzified some of these attributes, one can form one production rule depending on one or more measure. For example we have made the following rule depending on mean and variance of class Mo, such that if variance of attribute١ and attribute ٢ is less than some value, say ٠.٠٠١(small variance more effective attribute), we can make this rule:

IF $attrib_1$ is M_1 and $attrib_2$ is M_2 then class M_o Or

IF $attrib_1$ is M_1 and $attrib_2$ is M_2 then $attrib_x$ is M

Where: M_1, M_2 are the mean of attribute ١ and attribute ٢ respectively. *Mo* is the attribute of largest mode. See Appendix.

٤.١ Case Study

Because of the lack of the real database in the country establishments' information centers, we have visited the internet sites. These sites offer many types of databases attached with most needed prior information. We have taken five databases from different domains. The application areas they have covered are medical, cars evaluation, housing, and Ionosphere databases. Also, they vary in tables' dimensions. Some of them, like cars table, have a large number of records but a small number of attributes and vice versa.

Before running the system on each database, we have to collect some important prior information to it. This information can be used when some parameters need to enter to the system such as estimated max number of clusters and the categorical terms for some attributes like age (child, boy, young, old, ..,etc).

Four cases study used to test the DBRuleExtractor system. The prior information and results of each case are shown.

٤.١.١ Cancer Database

١. Title: Wisconsin Breast Cancer Database (January ٨, ١٩٩١)

٢. Sources:

-- Dr. WIlliam H. Wolberg (physician)

University of Wisconsin Hospitals

Madison, Wisconsin, USA

-- Donor: Olvi Mangasarian (mangasarian@cs.wisc.edu)

Received by David W. Aha (aha@cs.jhu.edu)

-- Date: ١٥ July ١٩٩٢

٣. Attributes:

Attributes ٢ through ١٠ have been used to represent instances. Each instance has one of ٢ possible classes: benign or malignant.

٤. Relevant Information:

Samples arrive periodically as Dr. Wolberg's reports of his clinical cases. The database therefore reflects this chronological grouping of the data.

٥. Number of Instances: ٦٩٩ (as of ١٥ July ١٩٩٢)

٦. Number of Attributes: ١٠ plus the class attribute

٧. Attribute Information: (class attribute has been moved to last column)

#Attribute	Domain
١. Sample code number	id number
٢. Clump Thickness	١- ١٠
٣. Uniformity of Cell Size	١ - ١٠
٤. Uniformity of Cell Shape	١ - ١٠
٥. Marginal Adhesion	١ - ١٠
٦. Single Epithelial Cell Size	١ - ١٠
٧. Bare Nuclei	١ - ١٠
٨. Bland Chromatin	١ - ١٠
٩. Normal Nucleoli	١ - ١٠
١٠. Mitoses	١ - ١٠
١١. Class:	(٢ for benign, ٤ for malignant)

٨. Missing attribute values: ١٦

There are ١٦ instances in Groups ١ to ٦ that contain a single missing (i.e., unavailable) attribute value, now denoted by "?".

٩. Class distribution:

Benign: ٤٥٨ (٦٥.٥٪)

Malignant: ٢٤١ (٣٤.٥٪)

Sample code number: continuous

Clump Thickness: continuous

Uniformity of Cell Size: continuous

Uniformity of Cell Shape: continuous

Marginal Adhesion: continuous

Single Epithelial Cell Size: continuous

Bare Nuclei: continuous

Bland Chromatin: continuous

Normal Nucleoli: continuous

Mitoses: continuous

This Text has been generated Using DBRuleExtractor System:

-Results of Step (١) Data Selection:

١-Databse Name: Cancer.mdb

٢-Table Name: BREAST

٣-Number of Selected Fields: ١٠

٤-Feilds selected are: Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, Mitoses, Class.

Results of Step (٢) Data Preprocessing & Normalization

Fields have to be coded are: Class.

All fields have been scaled to the range [-١:١]

All fields have been normalized to unit one vector

Results of Step (٣) Genetic Algorithm - Cluster Seeds Detection

Parameters:

Population Size is: ٣٢

Chromosome Size is: ١٦

Maximum Number of cluster expected: ٨
Minimum Number of cluster expected: ٢

Limit Of generation count: ٢٠٠

Number of detected clusters is: ٦

DBi:. ٠.٣٦٤٦٨١٥٥٥١

-Results of Step (٤) neural network clustering-training

Parameters:

Number of Nodes :(got from GA):٦

Max Training Steps: ١٥٠

Alpha Value: ٠.٥

Momentum Value: ٠.٨

Weight Initialization Method :(Centers Of clusters From GA/Randomly): centers of clusters From GA

-Result of Step (٥) Rules survey:

Statistical Analysis for each class....Done

Fuzzy mapping for qualitative attributes....Done

Forming IF..... Then Rules.....Done

The Target attribute is Class

Rules been generated for each class are as follow:

IF (Clump Thickness is ٨) **AND** (Uniformity of Cell Size is ٧) **AND** (Uniformity of Cell Shape is ٧) **AND** (Marginal Adhesion is ٥) **AND** (Single Epithelial Cell Size is ٥) **AND** (Bare Nuclei is ١٠) **AND** (Bland Chromatin is ٧) **AND** (Normal Nucleoli is ٦) **AND** (Mitoses is ١) **THEN** Class is MALIGNANT

IF (Clump Thickness = ٥) **AND** (Uniformity of Cell Size = ٨) **AND** (Uniformity of Cell Shape = ٧) **AND** (Marginal Adhesion = ٧) **AND** (Single Epithelial Cell Size = ٦) **AND** (Bare Nuclei = ٨) **AND** (Bland Chromatin = ٧) **AND** (Normal Nucleoli = ٧) **AND** (Mitoses = ٧) **THEN** Class is MALIGNANT

IF (Uniformity of Cell Size IS ١.٣٨٠٩٥) **AND** (Uniformity of Cell Shape IS ١.٤٦٩٣٩) **AND** (Marginal Adhesion IS ١.٣٨٧٧٦) **AND** (Single Epithelial Cell Size IS ٢.١٣٦٠٥) **AND** (Bare Nuclei IS ١.٤٠٨١٦) **AND** (Bland Chromatin IS ٢.١٠٥٤٤) **AND** (Normal Nucleoli IS ١.٣١٦٣٣) **AND** (Mitoses IS ١.٠٧٨٢٣) **THEN** Class is BENIGN

IF (Clump Thickness is ٣) **AND** (Uniformity of Cell Size is ١) **AND** (Uniformity of Cell Shape is ١) **AND** (Marginal Adhesion is ١) **AND** (Single Epithelial Cell Size is ٢) **AND** (Bare Nuclei is ١) **AND** (Bland Chromatin is ٢) **AND** (Normal Nucleoli is ١) **AND** (Mitoses is ١) **THEN** Class is BENIGN

IF (Clump Thickness = ٧) **AND** (Uniformity of Cell Size = ٧) **AND** (Uniformity of Cell Shape = ٧) **AND** (Marginal Adhesion = ٧) **AND** (Single Epithelial Cell Size = ٢) **AND** (Bare Nuclei = ٢) **AND** (Bland Chromatin = ٢) **AND** (Normal Nucleoli = ٢) **AND** (Mitoses = ٢) **THEN** Class is BENIGN

٤.١.٢ Heart Disease Databases

١. Title: Heart Disease Databases

٢. Source Information:

(a) Creators:

-- ١. Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.

-- ٢. University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.

-- ٣. University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.

-- ٤. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation:

Robert Detrano, M.D., Ph.D.

(b) Donor: David W. Aha (aha@ics.uci.edu) (٧١٤) ٨٥٦-٨٧٧٩

(c) Date: July, ١٩٨٨

٣. Relevant Information:

This database contains ٧٦ attributes, but all published experiments refer to using a subset of ١٤ of them. In particular, the Cleveland database is the only one that has been used by ML researchers to this date. The "goal" field refers to the presence of heart disease in the patient. It is integer valued from ٠ (no presence) to ٤.

The names and social security numbers of the patients have recently been removed from the database, replaced with dummy values. One file has been "processed", that one contains the Cleveland database. All four unprocessed files also exist in this directory.

٤. Number of Attributes: ١٣

٥. Attribute Information:

No	Attribute	Description
١.	Sick	Healthy=٠ Sick>٠
٢.	Male	Female=٠ Male=١
٣.	Age	Years
٤.	ChestPain	Asympt=٠ Notang=١ Abnang=٢ Angina=٣
٥.	BloodPres	Trestbps (resting blood pressure)
٦.	Cholesteral	
٧.	LowBloodSugar	False=٠ True=١ (fasting blood sugar<١٢٠)
٨.	ECG	Normal=٠ Abn=١ Hyper=٢
٩.	HeartRate	Maximum heart rate
١٠.	Angina	False=٠ True=١
١١.	OldPeak	
١٢.	Slope	Flat=٠ Down=-١ Up=١
١٣.	Vessel Count	Number of vessels colored
١٤.	Thal	Normal=٠ Rev=-١ Fix=١

٦. Missing Attribute Values: Several. Distinguished with value(-٩.٠)

٧. Results for this data base are as follow:

This Text has been generated Using DBRuleExtractor System

-Results of Step (١) Data Selection

١-Database Name: D:\PROJ-٢٧-٦\databases\Heart٢٠٠٠.mdb

٢-Table Name: Heartdata

٣-Number of Selected Fields: ١٤

٤-Feilds have been selected are: Status, Sex, Age, ChestPain, BloodPres, Cholesteral, LowBloodSugar, ECG, HeartRate, Angina, OldPeak, Slope, VesselCount, Thal.

-Results of Step (٢) Data Preprocessing & Normalization

Fields have to be coded are: Status, Sex, ChestPain, LowBloodSugar, ECG, Angina, Slope, Thal.

All fields have been scaled to the range [-١:١]

All fields have been normalized to unit one vector

-Results of Step (٣) Genetic Algorithm - Cluster Seeds Detection

Parameters:

Population Size is: ٣٢

Chromosome Size is: ١٦

Maximum Number of cluster expected: ٣

Minimum Number of cluster expected: ٢

Limit Of generation count: ١٥٠

Number of detected clusters is: ٢

DBi: ٠.٩٦٨٤٩٩٠٤

-Results of Step (٤) neural network clustering-training

Parameters:

Number of Nodes :(got from GA):٢

Max Training Steps: ١٠٠

Alpha Value: ٠.٥

Momentum Value: ٠.٨

Weight Initialization Method :(Centers Of clusters From GA/Randomly): centers of clusters From GA

Result of Step (٥) Rules survey:

Statistical Analysis for each class....Done

Fuzzy mapping for qualitative attributes....Done

Forming IF..... Then Rules.....Done

The Target attribute is Status

Rules been generated for each class are:

IF (Age IS ٥٦) **AND** (BloodPres IS ١٣٣) **AND** (Cholesteral IS ٢٤٧) **AND** (HeartRate IS ١٣٩) **THEN** Status is Sick

IF (Sex is male) **AND** (Age is ٥٨) **AND** (ChestPain is Abnang) **AND** (BloodPres is ١٣٠) **AND** (Cholesteral is ٢٤٦) **AND** (LowBloodSugar is false) **AND** (ECG is normal) **AND** (HeartRate is ١٤٢) **AND** (Angina is true) AND (OldPeak is ١.٤) **AND** (Slope is Up) **AND** (VesselCount is ١) **AND** (Thal is fix) **THEN** Status is Sick

IF (Age is between (٣٥ - ٧٧)) **AND** (BloodPres is between (٩٤ - ٢٠٠)) AND (Cholesteral is between (١٣١ - ٤٠٩)) AND (HeartRate is between (٧١ - ١٩٥)) THEN Status is Sick

IF (Age IS ٥٣) **AND** (BloodPres IS ١٣٠) **AND** (Cholesteral IS ٢٤٥) **AND** (HeartRate IS ١٥٩) **AND** (OldPeak IS .٥٧٩٧٥) **AND** (Thal IS Rev) **THEN** Status is healthy

IF (Sex is male) **AND** (Age is ٥٣) **AND** (ChestPain is Notang) **AND** (BloodPres is ١٣٠) **AND** (Cholesteral is ٢٣٩) **AND** (LowBloodSugar is false) AND (ECG is Hyper) **AND** (HeartRate is ١٦٢) **AND** (Angina is false) AND (OldPeak is .٢) **AND** (Slope is Flat) **AND** (VesselCount is ٠) **AND** (Thal is Rev) **THEN** Status is healthy

IF (Sex = female) **AND** (ChestPain = Notang) **AND** (BloodPres = ١٣٠) **AND** (LowBloodSugar = true) **AND** (ECG = Hyper) **AND** (Angina = false) **AND** (OldPeak = ٤.٦٦٥٣٥) **AND** (Slope = Flat) **AND** (VesselCount = ٢) **AND** (Thal = Rev) **THEN** Status is healthy

٤.١.٣ Car Marketing Database

١. Title: Car Evaluation Database

٢. Sources:

(a) Creator: Marko Bohanec

(b) Donors: Marko Bohanec (marko.bohanec@ijs.si)

Blaz Zupan (blaz.zupan@ijs.si)

(c) Date: June, ١٩٩٧

٣. Relevant Information Paragraph:

Car Evaluation Database has been derived from a simple hierarchical decision model originally developed for the demonstration of DEX (M.Bohanec, V. Rajkovic: Expert system for decision making. Sistemica ١(١),pp. ٤٥-١٥٧, ١٩٩٠.). The model evaluates cars according to the following concept structure:

CAR	car acceptability
. PRICE	overall price
. . buying	buying price
. . maint	price of the maintenance
. TECH	technical characteristics
. . COMFORT	comfort

. . . doors number of doors

. . . persons capacity in terms of persons to carry

. . . lug_boot the size of luggage boot

. . safety estimated safety of the car

Input attributes are printed in lower case. Besides the target concept (CAR), the model includes three intermediate concepts:

PRICE, TECH, COMFORT. Every concept is in the original model related to its lower level descendants by a set of examples (for these examples sets see http://www-ai.ijs.si/BlazZupan/car.html).

The Car Evaluation Database contains examples with the structural information removed, i.e., directly relates CAR to the six input attributes: buying, maint, doors, persons, lug_boot, safety.

Because of known underlying concept structure, this database may be particularly useful for testing constructive induction and structure discovery methods.

٤. Number of Instances: ١٧٢٨

(Instances completely cover the attribute space)

٥. Number of Attributes: ٦

٦. Attribute Values:

buying v-high, high, med, low

maint v-high, high, med, low

doors ٢, ٣, ٤, ٥-more

persons ٢, ٤, more

lug_boot small, med, big

safety low, med, high

٧. Missing Attribute Values: none

٨. Class Distribution (number of instances per class)

class N N[%]

---

unacc ١٢١٠ (٧٠.٠٢٣ %)

acc ٣٨٤ (٢٢.٢٢٢ %)

good ٦٩ (٣.٩٩٣ %)

v-good ٦٥ (٣.٧٦٢ %)

This text has been generated Using DBRuleExtractor System

-Results of Step (١) Data Selection

١-Databse Name: D:\PROJ-٢٧-٦\databases\Cars.mdb

٢-Table Name: cars

٣-Number of Selected Feilds: ٧

٤-Feilds have been selected are: buying, maint, doors, persons, lug_boot, safety, class,

-Results of Step (٢) Data Preprocessing & Normalization

Fields have to be coded are: buying, maint, doors, persons, lug_boot, safety, class,

All fields have been scaled to the range [-١:١]

All fields have been normalized to unit one vector

-Result of Step (٣) Genetic Algorithm - Cluster Seeds Detection

Parameters

Population Size is: ٣٢

Chromosome Size is: ١٦

Maximum Number of cluster expected: ١٦

Minimum Number of cluster expected: ٢

Limit Of generation count: ١٠

Number of detected clusters is: ٢

DBi :.٧٥٨٦٧٤٢٩٨٥

-Results of Step (٤) neural network clustering-training

Parameters:

Number Of Nodes :(got from GA) :٢

Max Training Steps: ١٠

Alpha Value :٣.٠٢٣٣١E-٠٣

Momentum Value :٠.٨

Weight Initialization Method :(Centers Of clusters From GA/Randomly): centers of clusters From GA

-Result of Step (٥) Rules survey:

Statistical Analysis for each class....Done

Fuzzy mapping for qualitative attributes....Done

Forming IF..... THEN Rules.....Done

The Target attribute is class

Rules generated for each class are:

IF (buying is med) **AND** (maint is med) **AND** (doors is ٣) **AND** (persons is ٤) **AND** (lug_boot is med) **AND** (safety is med) **THEN** class is acc

IF (buying = med) **AND** (maint = med) **AND** (doors = ٤) **AND** (persons = ٤) **AND** (lug_boot = med) **AND** (safety = low) **THEN** class is acc

---

IF (buying is low) **AND** (maint is low) **AND** (doors is ٤) **AND** (persons is ٤) **AND** (lug_boot is med) **AND** (safety is high) **THEN** class is good

IF (buying = low) **AND** (maint = low) **AND** (doors = ٤) **AND** (persons = ٤) **AND** (lug_boot = med) **AND** (safety = low) **THEN** class is good

٤.١.٤ Housing Database

١. Title: Boston Housing Data

٢. Sources:

(a) Origin: This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University.

(b) Creator: Harrison, D. and Rubinfeld, D.L. 'Hedonic prices and the demand for clean air', J. Environ. Economics & Management, vol.٥, ٨١-١٠٢, ١٩٧٨.

(c) Date: July ٧, ١٩٩٣.

٣. Past Usage:

- Used in Belsley, Kuh & Welsch, 'Regression diagnostics ...', Wiley, ١٩٨٠. N.B. Various transformations are used in the table on pages ٢٤٤-٢٦١.

- Quinlan,R. (١٩٩٣). Combining Instance-Based and Model-Based Learning. In Proceedings on the Tenth International Conference of Machine Learning, ٢٣٦-٢٤٣, University of Massachusetts, Amherst. Morgan Kaufmann.

٤. Relevant Information:

Concerns housing values in suburbs of Boston.

٥. Number of Instances: ٥٠٦

٦. Number of Attributes: ١٣ continuous attributes (including "class" attribute "MEDV"), ١ binary-valued attribute.

٧. Attribute Information:

١. CRIM per capita crime rate by town.
٢. ZN proportion of residential land zoned for lots over ٢٥,٠٠٠ sq. ft.
٣. INDUS proportion of non-retail business acres per town
٤. CHAS Charles River dummy variable (= ١ if tract bounds river; ٠ otherwise)
٥. NOX nitric oxides concentration (parts per ١٠ million)
٦. RM average number of rooms per dwelling.
٧. AGE proportion of owner-occupied units built prior to ١٩٤٠.
٨. DIS weighted distances to five Boston employment centers.
٩. RAD index of accessibility to radial highways.
١٠. TAX full-value property-tax rate per $١٠,٠٠٠.
١١. PTRATIO pupil-teacher ratio by town
١٢. B ١٠٠٠(Bk - ٠.٦٣)^٢ where Bk is the proportion of blacks by town
١٣. LSTAT % lower status of the population
١٤. MEDV Median value of owner-occupied homes in $١٠٠٠'s

٨. Missing Attribute Values: None.

This Text has been generated Using RulesExtractor System

Results of Step (١) Data Selection
١-Databse Name: F:\databases\Houses.mdb
٢-Table Name: HousePrices
٣-Number of Selected Fields: ١٤

٤-Feilds have been selected are: HousePrice, TaxRate, HouseSize, HouseAge, LotSize, RiverSide, CrimeRate, Industrial, AirQuality, Distance, Highways, Pupils/Teacher, Blacks, Poverty,

Results of Step (٢) Data Preprocessing & Normalization

Fields have to be coded are: RiverSide,

All fields have been scaled to the range [-١:١]

All fields have been normalized to unit one vector

Result of Step (٣) Genetic Algorithm - Cluster Seeds Detection parameters

Population Size is: ٣٢

Chromosome Size is: ١٦

Maximum Number of cluster expected: ٨

Minimum Number of cluster expected: ٢

Limit Of generation count: ٥٠٠

Number of detected clusters is: ٢

DBi :٠.٥٢٩٩٣٠١٤٣٦

Results of Step (٤) neural network clustering-training

Parameters:

Number of Nodes :(got from GA):٢

Max Training Steps: ١٠

Alpha Value: ٣.٠٢٣٣١E-٠٣

Momentum Value: ٠.٨

Weight Initialization Method :(Centers Of clusters From GA/Randomly): centers of clusters From GA

Result of Step (٥) Rules survey:

Statistical Analysis for each class....Done

Fuzzy mapping for qualitative attributes....Done

Forming (IF..... THEN)Rules.....Done

The Target attribute is Poverty

Rules have been generated for each class are:

IF (TaxRate IS ٦٣٨) **AND** (HouseSize IS ٥.٩٥٢١١) **AND** (HouseAge IS ٩٠.٦٩٠٩٧) **AND** (LotSize IS ٠) **AND** (CrimeRate IS ١١.١١٢٦٧) **AND** (Industrial IS ١٨.٦٤٢٤٥) **AND** (AirQuality IS .٦٨٧٠٦) **AND** (Distance IS ٢.٠٠٠٧٥) **AND** (Pupils/Teacher IS ١٩.٧٢٥٨١) **THEN** Poverty is ١٩.٠٢٨١٣

IF (HousePrice is ١٤.٦) **AND** (TaxRate is ٦٦٦) **AND** (HouseSize is ٦.١٠٣) **AND** (HouseAge is ٩٥) **AND** (LotSize is ٠) **AND** (RiverSide is ٠) AND (CrimeRate is ٧.٩٩٢٤٨) **AND** (Industrial is ١٨.١) **AND** (AirQuality is .٦٩٣) **AND** (Distance is ١.٨٦٨١) **AND** (Highways is ٢٤) **AND** (Pupils/Teacher is ٢٠.٢) **AND** (Blacks is ٣٧٢.٩٢) **THEN** Poverty is ١٩.٠٢٨١٣

IF (TaxRate = ٦٦٦) **AND** (RiverSide = ٠) **AND** (Industrial = ٤.٤٦٨٨) **AND** (AirQuality = .٧١٣) **AND** (Highways = ٥) **AND** (Pupils/Teacher = ٢٠.٢) **THEN** Poverty is ١٩.٠٢٨١٣

IF (TaxRate is between (٤٠٣ - ٧١١)) **AND** (HouseSize is between (٣.٥٦١ - ٨.٧٨)) **AND** (HouseAge is between (٤٠.٣ - ١٠٠)) **AND** (CrimeRate is between (.١٠٥٧٤ - ٨٨.٩٧٦٢)) **AND** (Industrial is between (١٨.١ - ٢٧.٧٤)) **AND** (AirQuality is between (.٥٣٢ - .٨٧١)) **AND** (Distance is between (١.١٢٩٦ - ٤.٠٩٨٣)) **AND** (Pupils/Teacher is between (١٤.٧ - ٢١.٢)) **THEN** Poverty is ١٩.٠٢٨١٣

IF (TaxRate IS ٣٠٧) **AND** (HouseSize IS ٦.٤٣١٤٨) **AND** (CrimeRate IS .٣٠١٩٣) **AND** (AirQuality IS .٤٩٦٢٤) **AND** (Highways IS ٤) **AND** (Pupils/Teacher IS ١٧.٨٩٤٥٩) **AND** (Blacks IS ٣٨٤.٨٦٣٢٨) **THEN** Poverty is ٩.٨٣٧٨٦.

IF (HousePrice is ٢٣.١) **AND** (TaxRate is ٣٠٤) **AND** (HouseSize is ٦.٣١) **AND** (HouseAge is ٥٩.٦) **AND** (LotSize is ٠) **AND** (RiverSide is ٠) **AND** (CrimeRate is .١٢٦٥) **AND** (Industrial is ٦.٢) **AND** (AirQuality is .٤٨٩) **AND** (Distance is ٤.٢٣٣) **AND** (Highways is ٤) **AND** (Pupils/Teacher is ١٨.٢) **AND** (Blacks is ٣٩٢.٩) **THEN** Poverty is ٩.٨٣٧٨٦

IF (LotSize = ٢١.٤) **AND** (RiverSide = ١) **AND** (Highways = ٤) **THEN** Poverty is ٩.٨٣٧٨٦

IF (TaxRate is between (١٨٧ - ٤٦٩)) **AND** (HouseSize is between (٤.٩٧٣ - ٨.٧٢٥)) **AND** (CrimeRate is between (.٠٠٦٣٢ - ٢.٩٢٤)) **AND** (AirQuality is between (.٣٨٥ - .٨٧١)) **AND** (Highways is between (١ - ٨)) **AND** (Pupils/Teacher is between (١٢.٦ - ٢٢)) **AND** (Blacks is between (٧٠.٨ - ٣٩٦.٩)) **THEN** Poverty is ٩.٨٣٧٨٦

٤.٢ Conclusions

The soft computing technique for data mining problem proposed in this thesis succeeds in:

١- There is no worst time since GA based clustering gives Winner-take-all network a suitable initialization parameters, this obvious from the fast convergence of this network meanly ١٠ to ٥٠ steps.
٢- Extracting the important components easy through running same dataset patterns frequently on the system.
٣- Extracting the important values of important components can be used to describe the clusters formed by production rules.
٤- Individual component boundaries could consider as fuzzy set are then used to extrapolate values that form the basis of rules.
٥- Fuzzy sets make these simple rules can be easily exploited by an expert or decision support system and are easily interpretable by an expert.
٦- The model has the ability to deal with any database in any application domain, this mean it is universal.
٧- The system takes a reasonable time when to mine rules.

٤.٣ Future Works

١- Developing the DBRulesExtractor system to deal with data of different format such as multimedia.
٢- Use navigation agents to collect data from different sources such as web.

Appendix A

DBRuleExtractor System screen shots

١-Data Selection Screen Shot

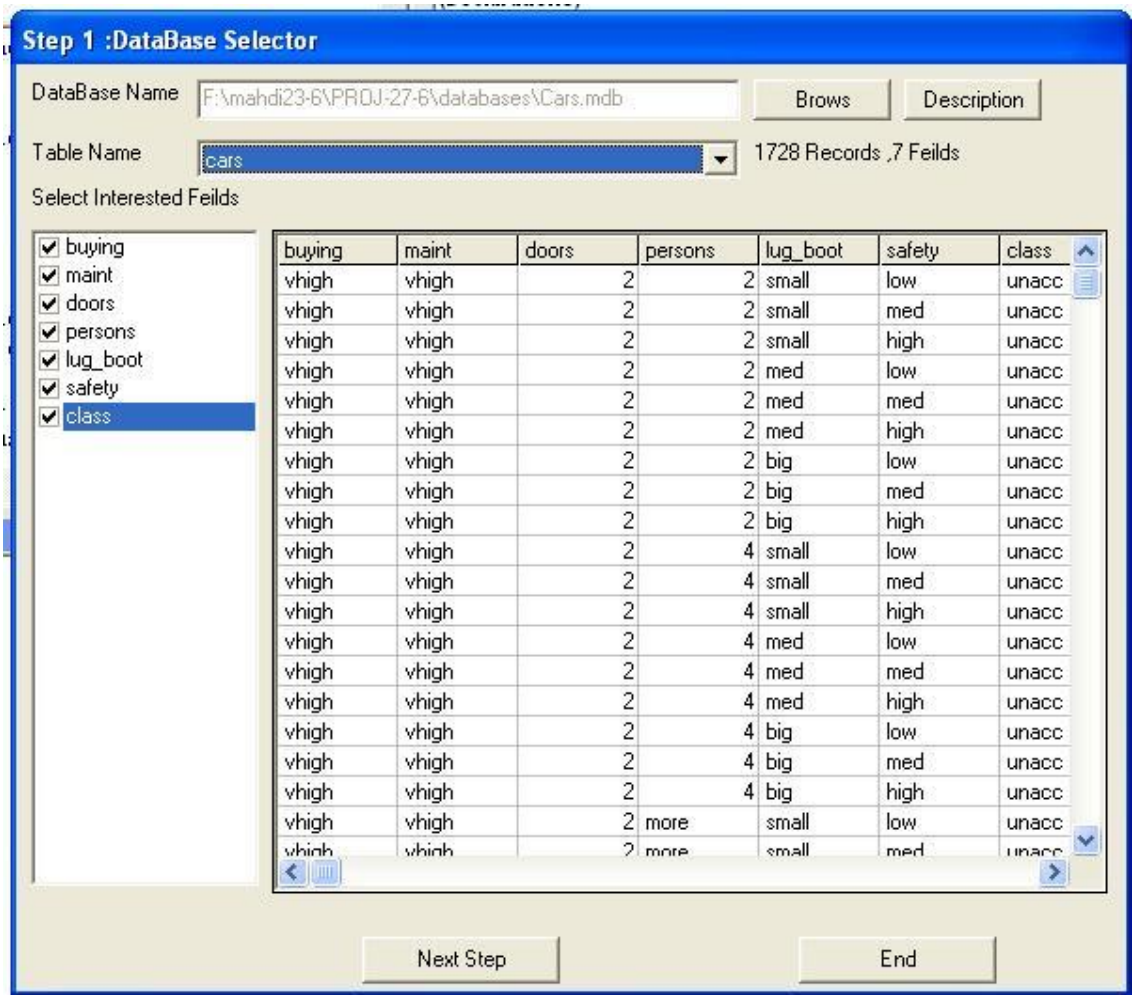

Figure (A-١) Attributes Selector Interface

٢- Preprocessing Screen Shot.

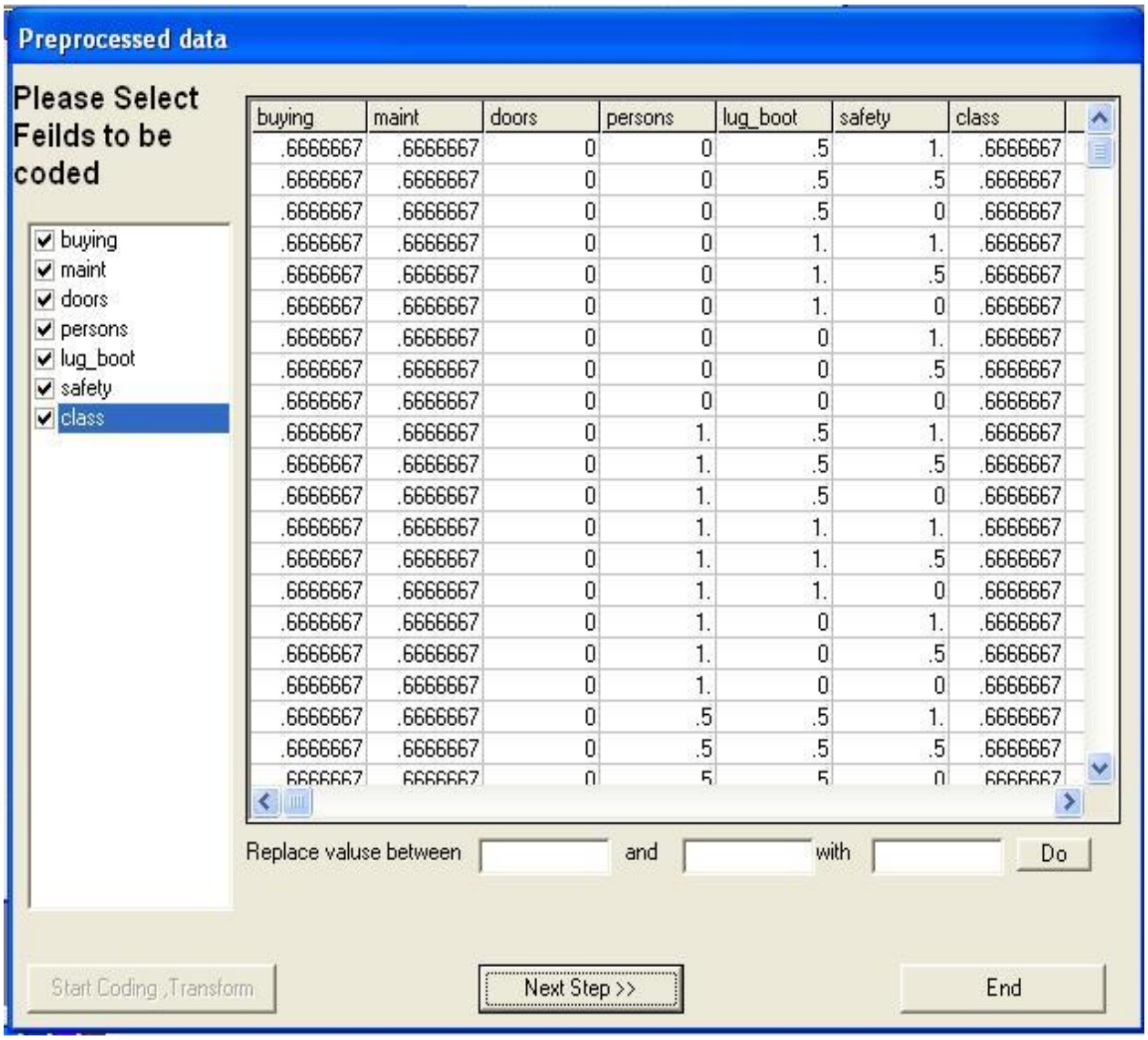

Figure (A-٢) Preprocessing Interface

٣- Clusters Seeds Detection Screen Shot

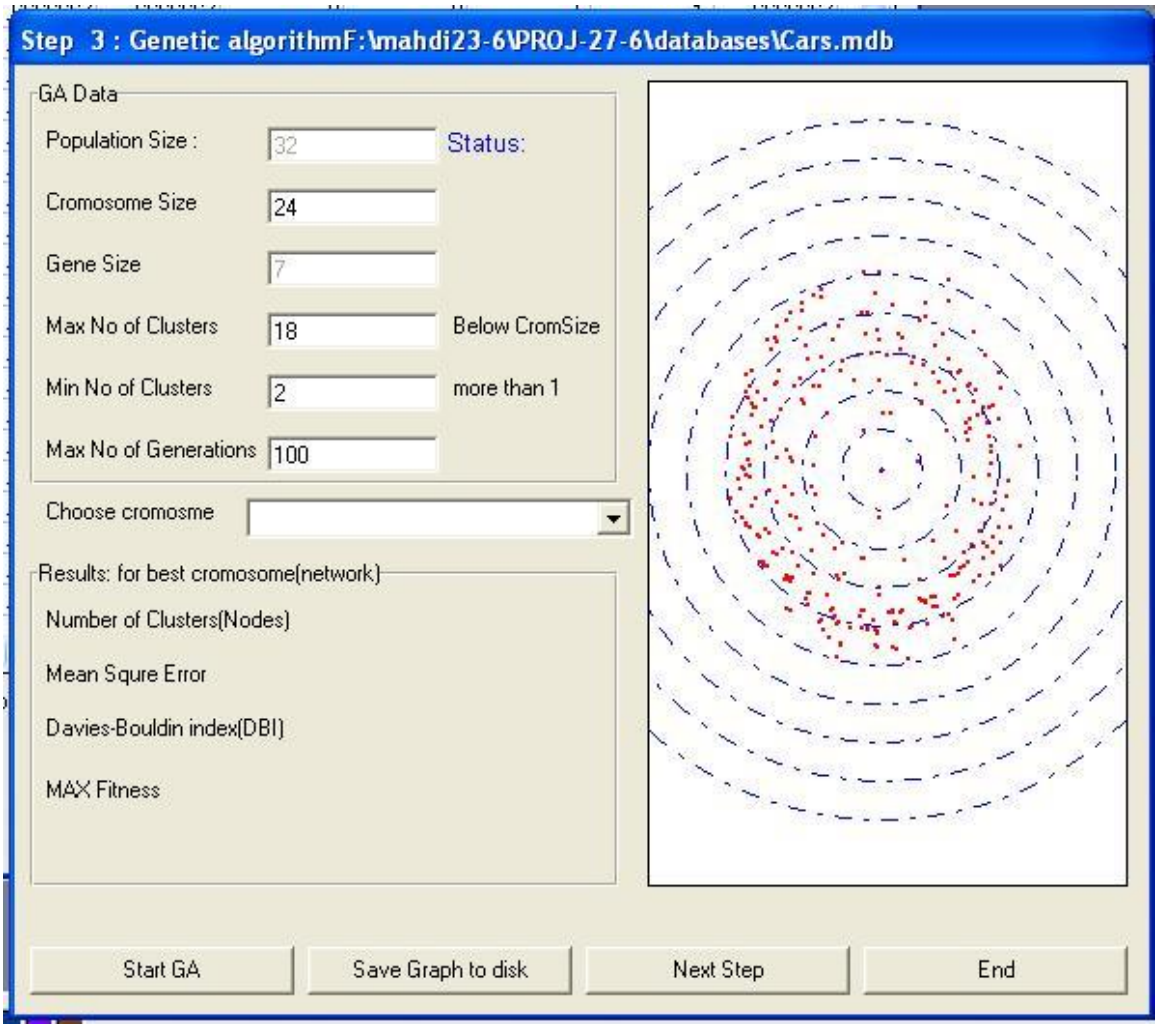

Figure (A-٣) Genetic Algorithm interface

٤- Result of GA Seeds detection

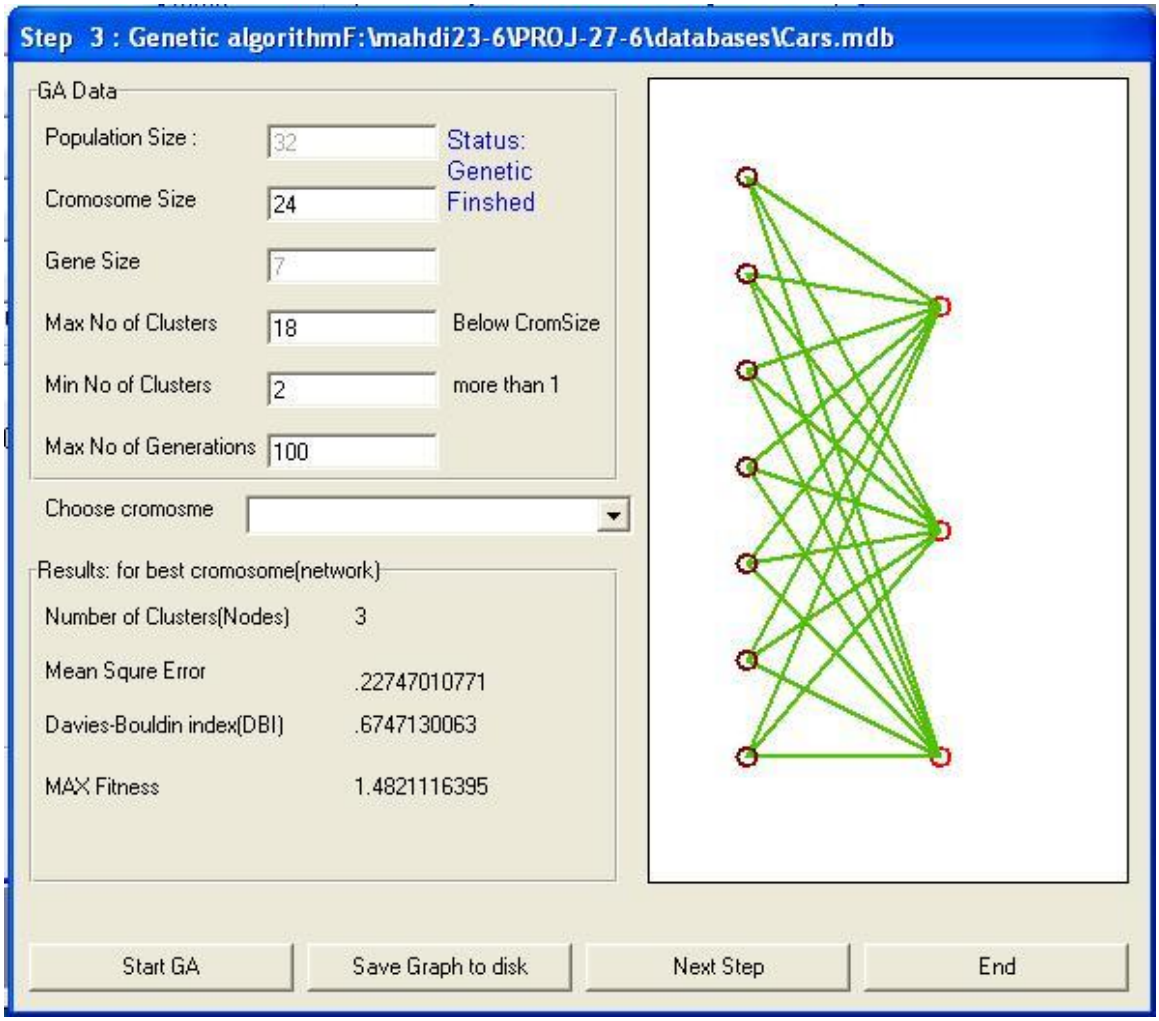

Figure (A-٤) Building ANN interface

٥- Clustering using Winner-Take-All screen shot

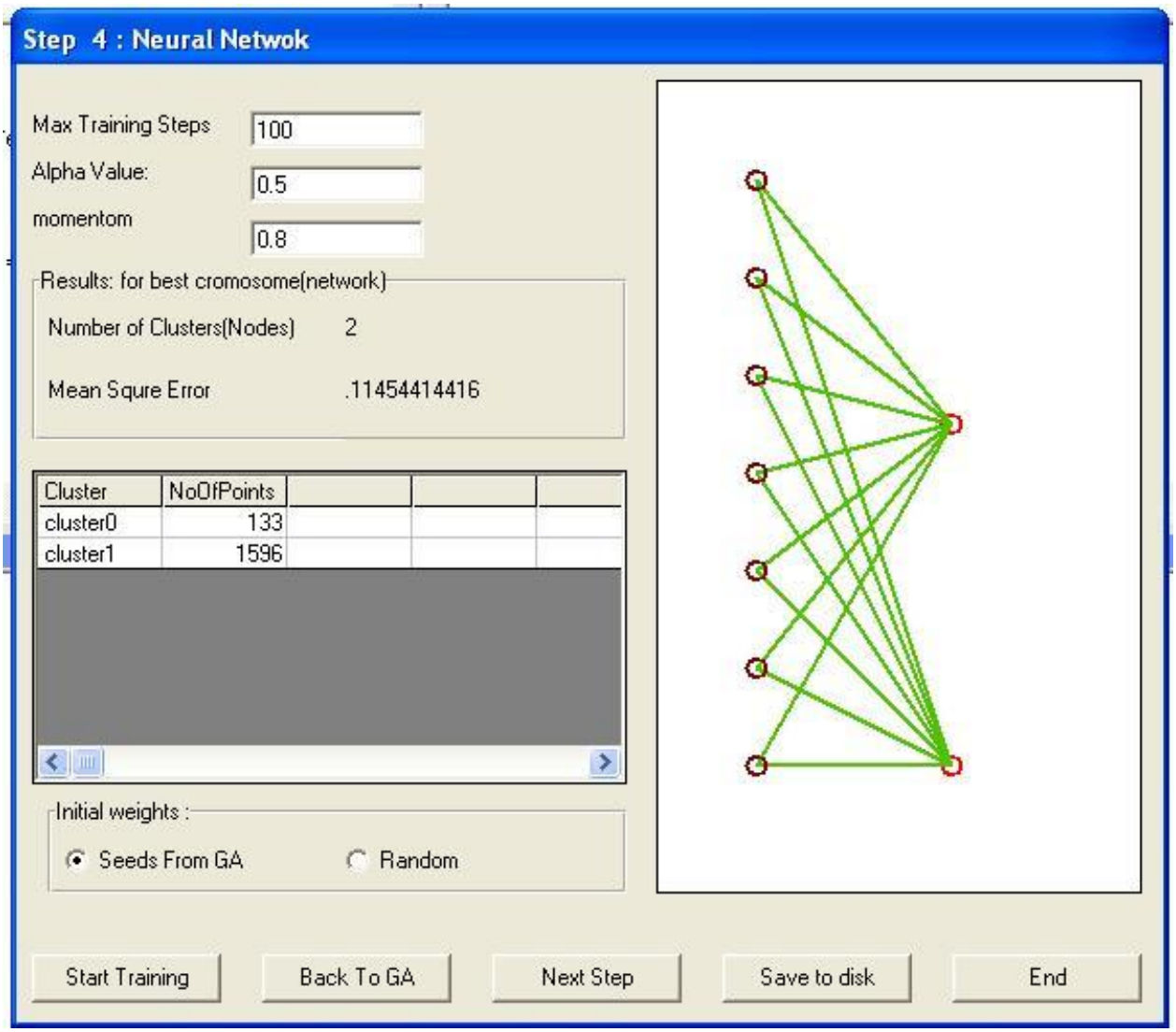

Figure (A-٥) Winner-Take-All learning

٦- Classification, Labeling Classes, and Rule Generation Screen Shot

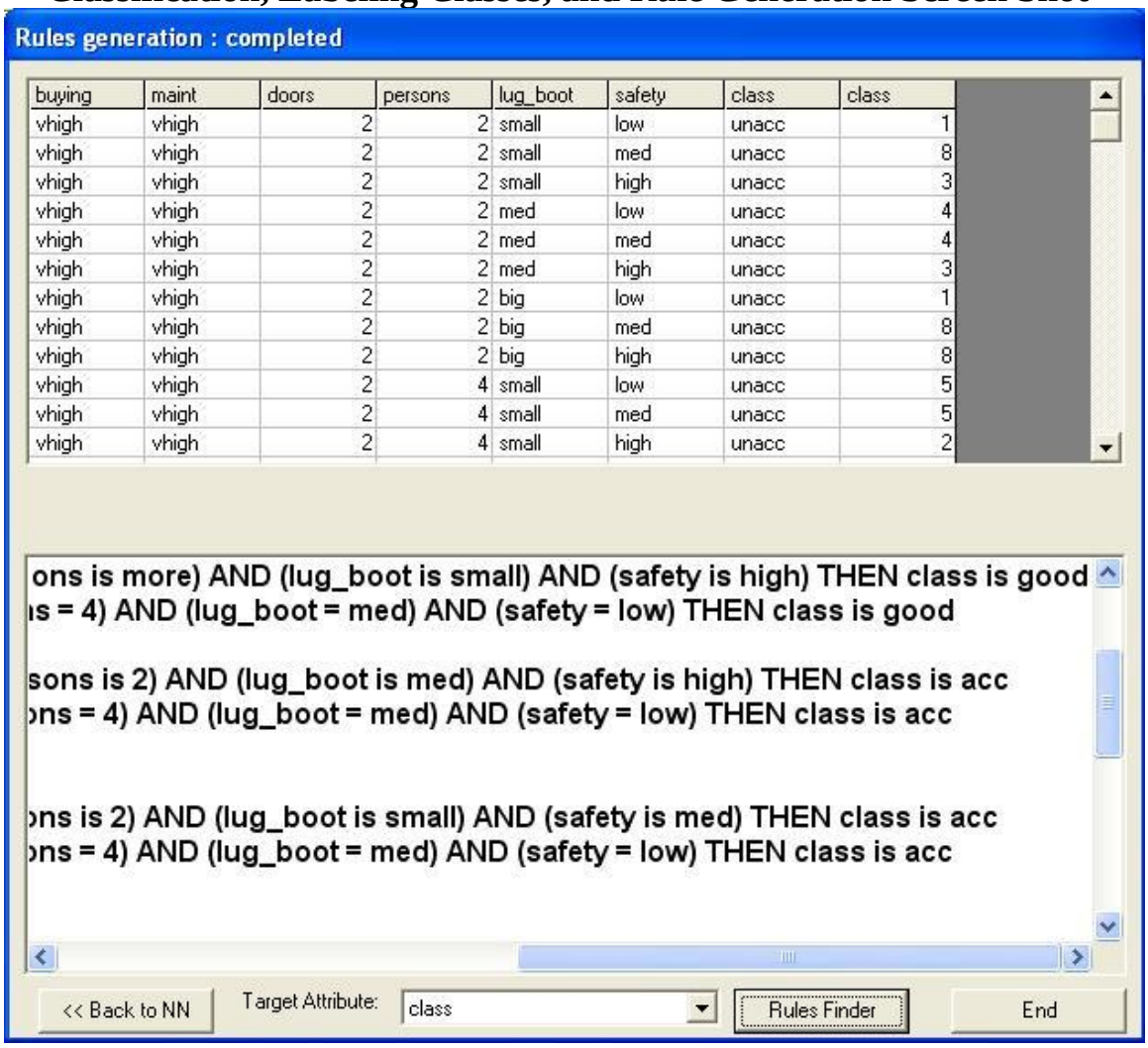

Figure (A-٦) Classification, Labeling Classes, and Rule Generation

٧-Data Selection Screen Shot

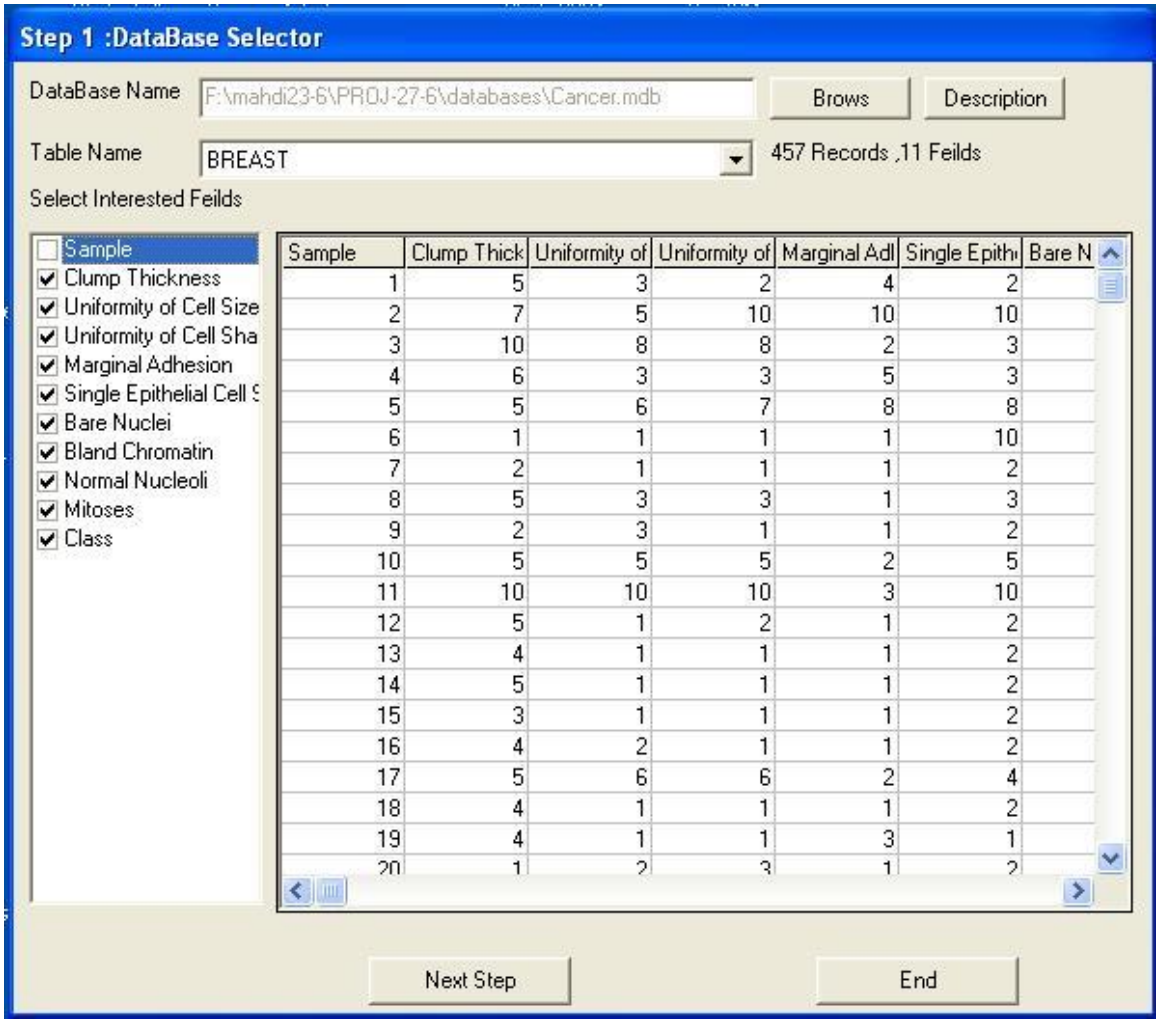

Figure (A-٧) Attributes Selector Interface

ʌ- Preprocessing Screen Shot.

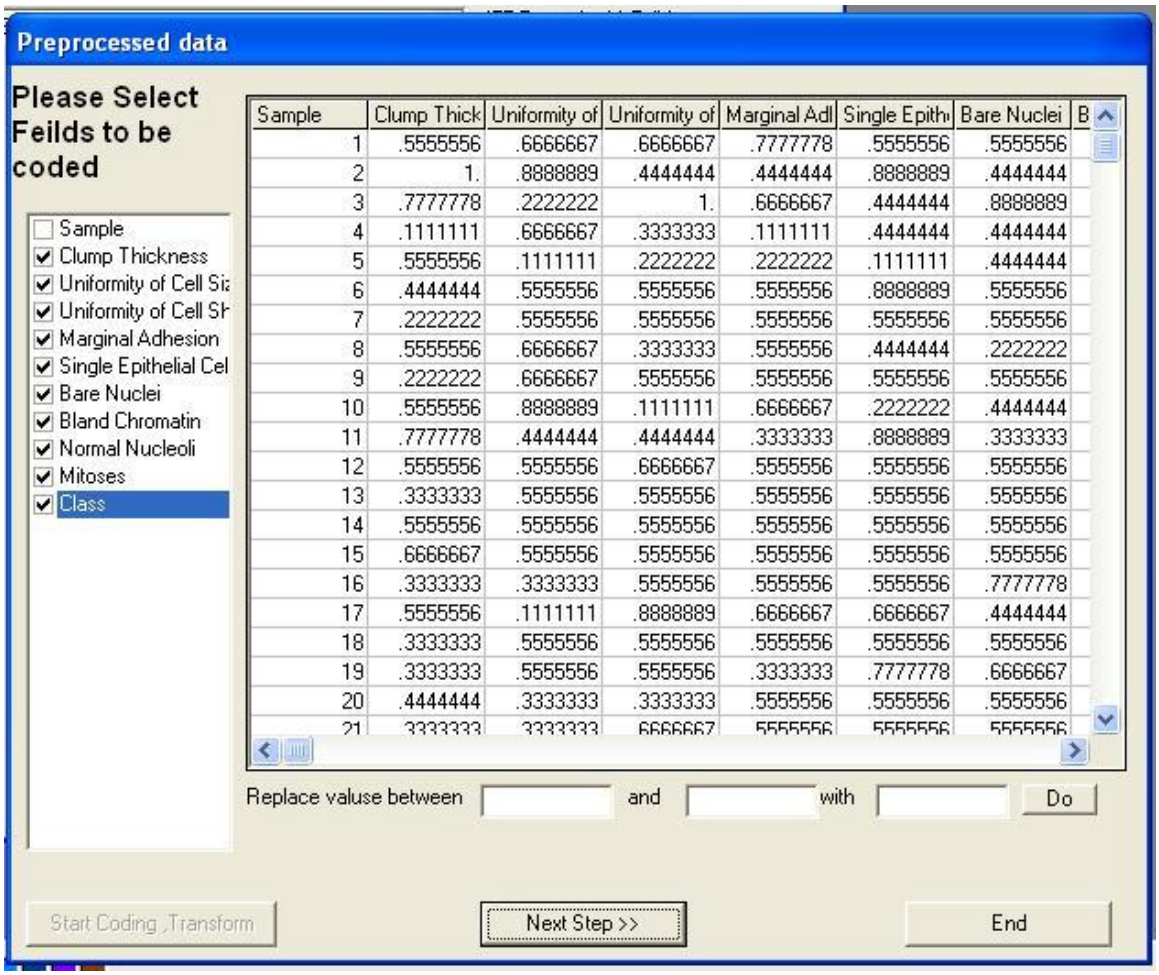

Figure (A-ʌ) Preprocessing Interface

٩- Clusters Seeds Detection Screen Shot

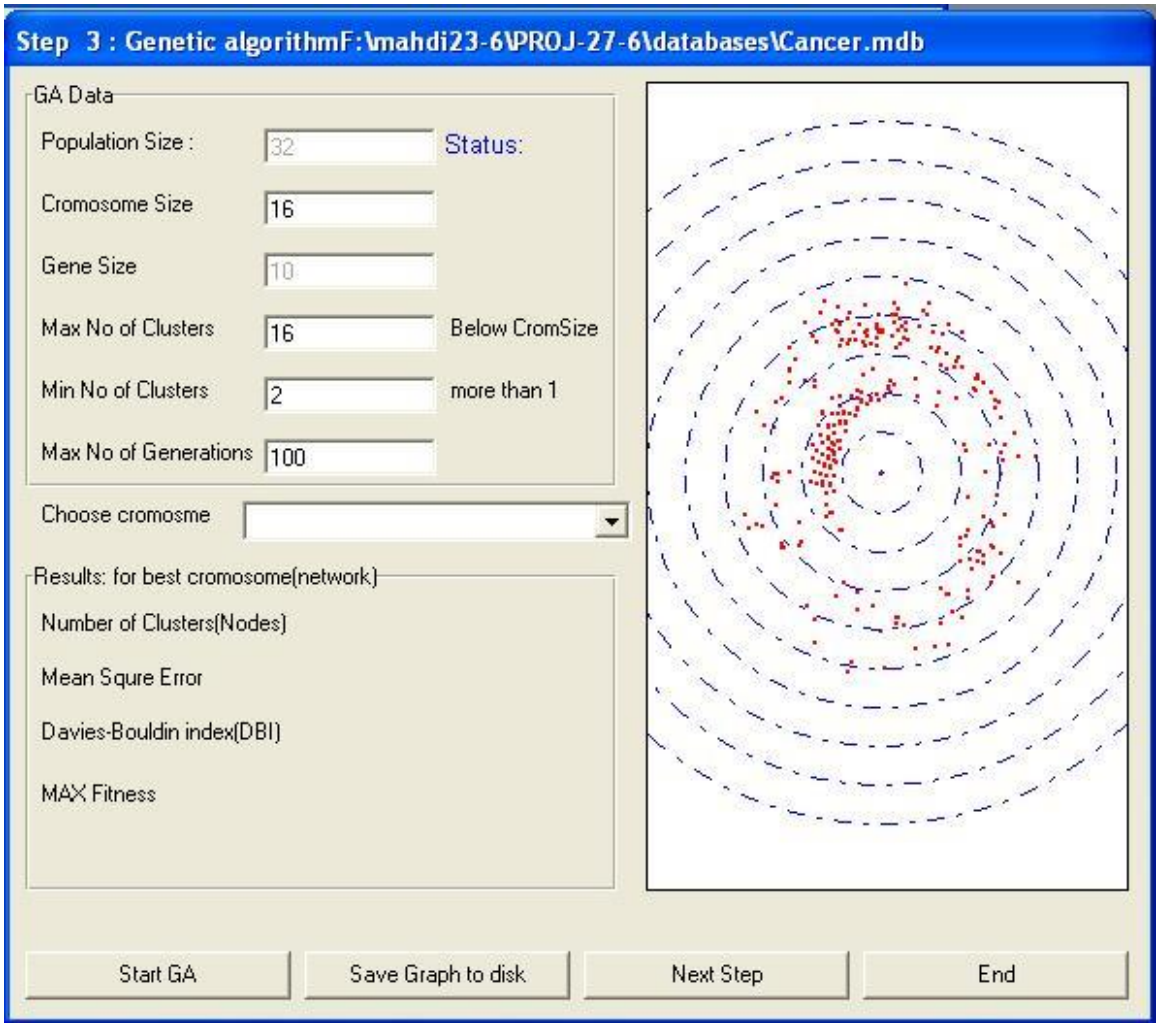

Figure (A-٩) Genetic Algorithm interface

١٠- Result of GA Seeds detection

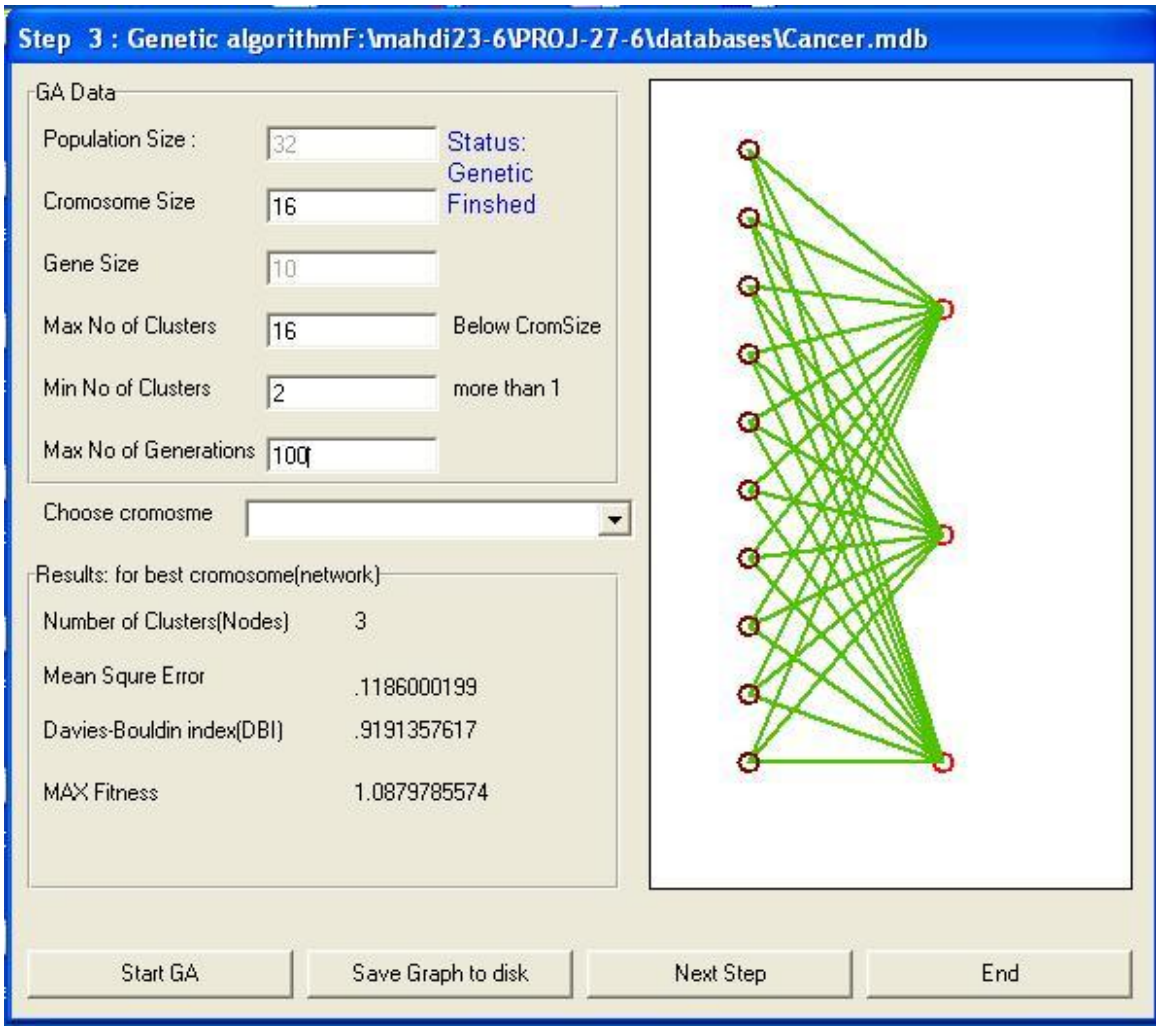

Figure (A-١٠) Building ANN interface

١١- Clustering using Winner-Take-All screen shot

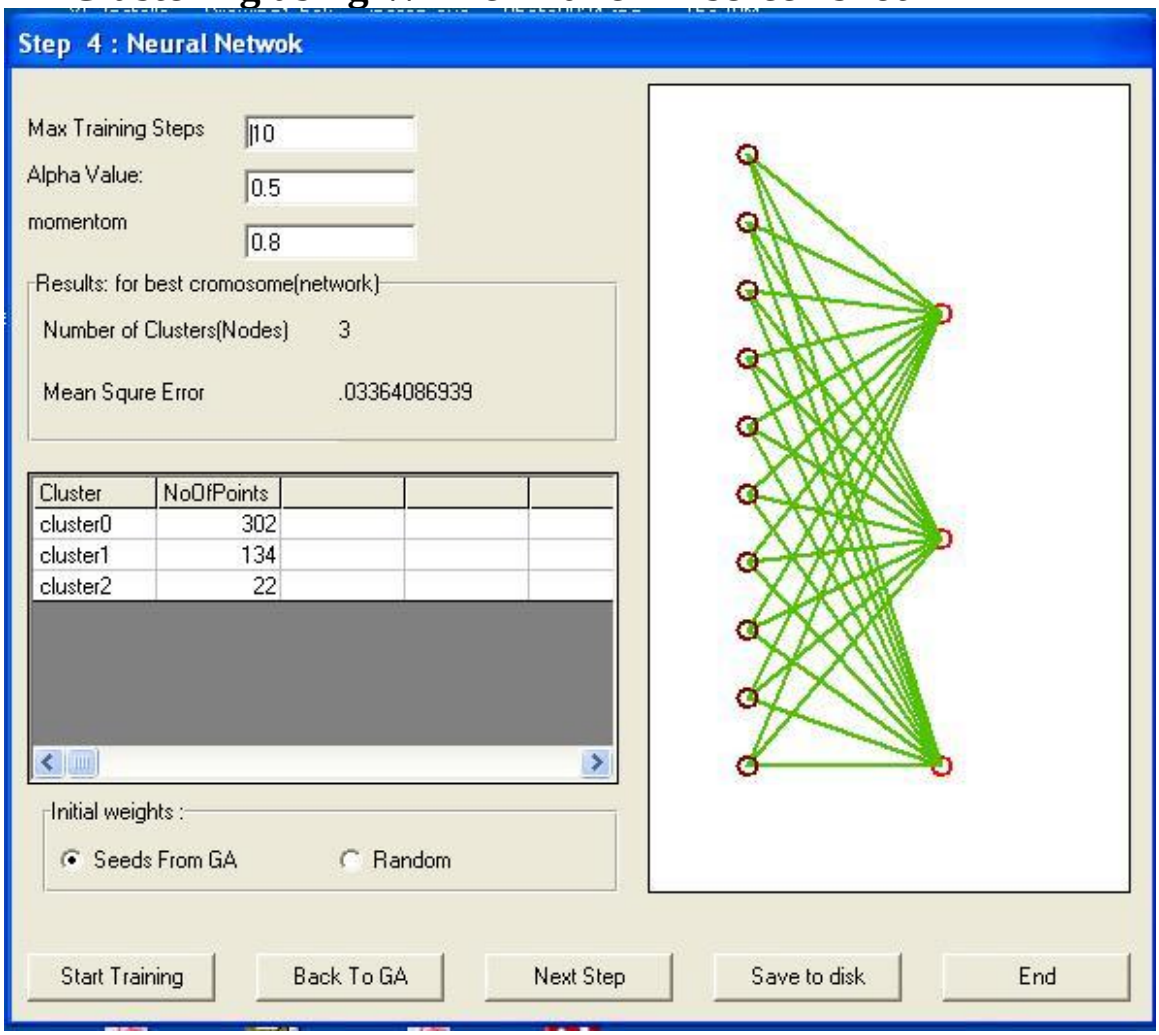

Figure (A-١١) Winner-Take-All learning

١٢- Classification, Labeling Classes, and Rule Generation Screen Shot

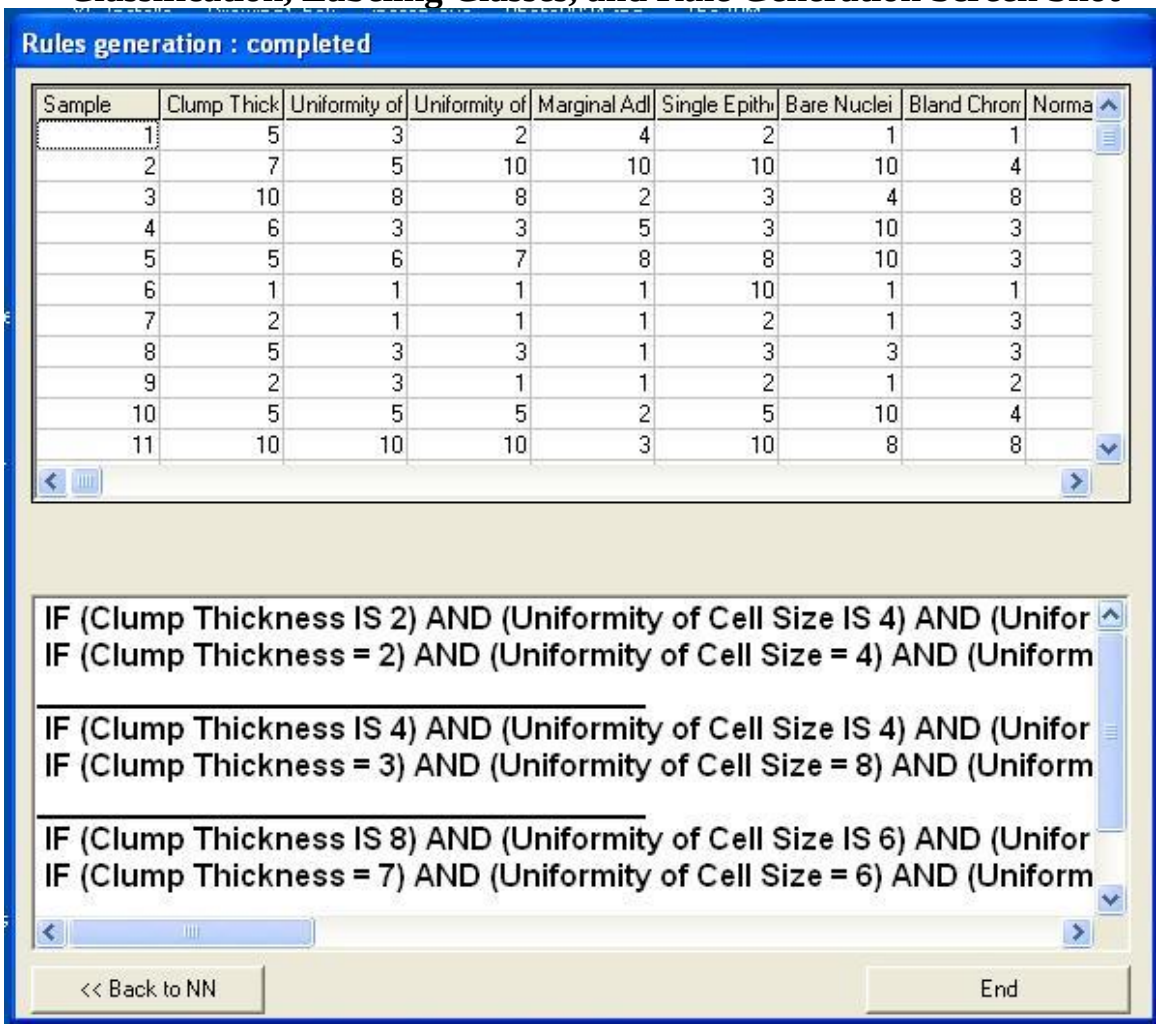

Figure (A-١٢) Classification, Labeling Classes, and Rule Generation

REFERENCES

[١] A. Konar, "**Artificial Inelegant and Soft Computing: Behavioral and Cognitive of the Human Brain,"** CRC Press, Florida, **٢٠٠٠.**

[٢] A.A. Freitas. "**A genetic algorithm for generalized rule induction,"** In: R. Roy et al. Advances in Soft Computing - Engineering Design and Manufacturing, ٣٤٠-٣٥٣. Springer-Verlag, ١٩٩٩.

[٣] A.A. Freitas. "**A Review of Evolutionary Algorithms for E-Commerce,"**. J. Segovia, P.S. Szczepaniak, M. Niedzwiedzinski (Eds.) E-Commerce and Intelligent Methods. Studies in Fuzziness and Soft Computing, Vol. ١٠٥. VIII. Heidelberg: Springer-Verlag, ٢٠٠٢

[٤] A.A. Freitas. "**A survey of evolutionary algorithms for data mining and knowledge discovery,"**. Springer-Verlag, ٢٠٠٢

[٥] A.A. Freitas. "**Book Review: Data mining using grammar-based genetic programming and applications",**. Genetic Programming and Evolvable Machines, ٢(٢), ١٩٧-١٩٩. June ٢٠٠١.

[٦] A.A. Freitas. "**Understanding the Crucial Role of Attribute Interaction in Data Mining**. Artificial Intelligence Review ١٦(٣), Nov. ٢٠٠١, pp. ١٧٧-١٩٩.

[٧] A.K. Jain; M.N. Murty; and P.J. Flynn, "**Data Clustering: A Review**", ACM Computing Surveys, Vol. ٣١, No. ٣, September ١٩٩٩.

[٨] Addison D., Wermter S., MacIntyre J. "**Effectiveness of Feature Extraction in Neural Network Architectures for Novelty Detection**," *Proceedings of the International Conference on Artificial Neural Networks*. p. ٩٧٦-٩٨١, Edinburgh, UK, September ١٩٩٩

[٩] Bandyopadhyay, Sanghamitra and Ujjwal Maulik, "**Genetic clustering for automatic evolution of clusters and application to image classification**", Pattern Recognition, volume ٣٥ (٢٠٠٢), number ٦, pp. ١١٩٧-١٢٠٨

[١٠] C.C. Bojarczuk, H.S. Lopes, A.A. Freitas. "**Genetic programming for knowledge discovery in chest pain diagnosis,"**. IEEE Engineering in Medicine and Biology magazine - special issue on data mining and knowledge discovery, ١٩(٤), ٣٨-٤٤, July/Aug. ٢٠٠٠.

[١١] C.E. Bojarczuk, H.S. Lopes and A.A. Freitas. "**Data mining with constrained-syntax genetic programming: applications in medical data sets,"**. Intelligent Data Analysis in Medicine and Pharmacology, a Workshop at Medinfo-٢٠٠١. London, UK, Sep. ٢٠٠١.

[١٢] C.E. Bojarczuk, H.S. Lopes and A.A. Freitas. "**Discovering comprehensible classification rules using genetic programming: a case study in a medical domain**. Genetic and Evolutionary Computation Conference ٩٥٣-٩٥٨. Orlando, FL, USA, July ١٩٩٩.

[١٣] C.S. Fertig, A.A. Freitas, L.V.R. Arruda and C. Kaestner. "**A Fuzzy Beam-Search Rule Induction Algorithm. Principles of Data Mining and Knowledge Discovery,"**: ٣rd European Conf. Lecture Notes in Artificial Intelligence ١٧٠٤, ٣٤١-٣٤٧. Springer-Verlag, ١٩٩٩.

[١٤] D.L.A. Araujo, H.S. Lopes and A.A. Freitas. "**Rule discovery with a parallel genetic algorithm,"**. ٢٠٠٠ Genetic and Evolutionary Computation Workshop Program, ٨٩-٩٢. Las Vegas, NV, USA. July ٢٠٠٠.

[١٥] D.L.A. Araujo, H.S. Lopes, A.A. Freitas. "**A parallel genetic algorithm for rule discovery in large databases,"** ١٩٩٩ IEEE Systems, Man and Cybernetics Conf., v. III, ٩٤٠-٩٤٥. Tokyo, Oct. ١٩٩٩.

[١٦] D.R. Carvalho and A.A. Freitas. "**A genetic algorithm with sequential niching for discovering small-disjunct rules,"** Genetic and Evolutionary Computation Conf. New York, July ٢٠٠٢.

[١٧] D.R. Carvalho and A.A. Freitas. "A **genetic algorithm-based solution for the problem of small disjuncts. Principles of Data Mining and Knowledge Discovery,"** Lecture Notes in Artificial Intelligence ١٩١٠, ٣٤٥-٣٥٢. Springer-Verlag, ٢٠٠٠.

[١٨] Daniel S. Keen "**A Fuzzy Genetic Active Suspension Implementation**," a report, May ١٤, ٢٠٠٤

[١٩] Deborah R. Carvalho, Alex A. Freitas, "**A Genetic Algorithm for Discovering Small-Disjunct Rules in Data Mining**," ٢٠٠٢

[٢٠] Del Jesus Etal.: "**Induction Of Fuzzy-Rule-Based Classifiers**" IEEE Transactions On Fuzzy Systems, Vol. ١٢, No. ٣, June ٢٠٠٤.

[٢١] Demiriz, A., Bennett, K. P., and Embrechts, M. "**A Genetic Algorithm Approach for Semi-Supervised Clustering**," Smart Engineering Systems Design, ٢٠٠٢, Vol. ٤, pp ٣٥-٤٤.

[٢٢] E. Noda, A.A. Freitas, H.S. Lopes. "**Discovering interesting prediction rules with a genetic algorithm,"** Congress on Evolutionary Computation , ١٣٢٢-١٣٢٩. Washington D.C., USA, July ١٩٩٩.

[٢٣] Erhard Rahm; Hong Hai Do, "**Data Cleaning: Problems and Current Approaches**," Microsoft Research, Redmond, WA.,٢٠٠٠

[٢٤] H. Abbass, R. Sarker, C. Newton. "**Data Mining: a Heuristic Approach,"** pp. ١٩١-٢٠٨. London: Idea Group Publishing, ٢٠٠٢

[٢٥] J. Larocca Neto; A.D. Santos, C.A.A. Kaestner, A.A. Freitas. "**The integrated data mining tool MineKit and a case study of its application on video shop data,"** ٢nd Int. ICSC Symp. on Engineering of Intelligent Systems (EIS-٢٠٠٠). Scotland, July ٢٠٠٠.

[٢٦] J. M. Zurada, "**Introduction To Artificial Neural Systems**," ST. Paul, MN: West publishing company, ١٩٩٢.

[٢٧] J. Vesanto and E. Alhoniemi, "**Clustering of the self-organizing map**," IEEE Trans. Neural Networks, vol. ١١, pp. ٥٨٦–٦٠٠, ٢٠٠٠.

[٢٨] Jeffrey W. Seifert, "**Data Mining: An Overview,"** CRS Report for Congress, ٢٠٠٤.

[٢٩] L. A. Zadeh, "**Fuzzy logic, neural networks, and soft computing**," Commun. ACM, vol. ٣٧, pp. ٧٧–٨٤, ١٩٩٤.

[٣٠] M.V. Fidelis, H.S. Lopes and A.A. Freitas. "**Discovering comprehensible classification rules with a genetic algorithm,"** Congress on Evolutionary Computation - ٢٠٠٠ (CEC-٢٠٠٠), ٨٠٥-٨١٠. La Jolla, CA, USA, July/٢٠٠٠

[٣١] Malone J., McGarry K, Bowerman C., Wermter S. "**Rule Extraction from Kohonen Neural Networks. Automated Trend Analysis of Proteomics Data Using Intelligent Data Mining Architecture,**" Neural Computing Applications Journal, ٢٠٠٥

[٣٢] Malone, J., McGarry, K. and Bowerman, C., "**Using an Adaptive Fuzzy Logic System to Optimise Knowledge Discovery in Proteomics**," International Conference on Recent Advances in Soft Computing, November ٢٠٠٤, pp. ٨٠-٨٥

[٣٣] McGarry K. "**The Analysis of Rules Discovered by the Data Mining Process**," *٤th International Conference on Recent Advances in Soft Computing*, Nottingham, UK, December ٢٠٠٢

[٣٤] McGarry K., Tait J., Wermter S., MacIntyre J. "**Rule Extraction from Radial Basis Function Networks**," Proceedings of the International Conference on Artificial Neural Networks. p. ٦١٣-٦١٨, Edinburgh, UK, September ١٩٩٩.

[٣٥] Mitra and Hayashi: "**Neuro-Fuzzy Rule Generation: Survey In Soft Computing Framework**," IEEE Transactions On Neural Networks, Vol. ١١, No. ٣, May ٢٠٠٠.

[٣٦] Müller, J.-A. : "**Self-Organising Data Mining**. ICIM ٢٠٠٢. Lvov ٢٠٠٢.

[٣٧] Müller, J.-A., F.Lemke: "**Self-Organising Data Mining,"** BoD Hamburg ٢٠٠٠.

[٣٨] P. Berkhin, "**Survey of Clustering Data Mining Techniques**," *Accrue Software, Inc.*, ٢٠٠٢.

[٣٩] P. Mitra, S. Mitra, and K. Pal Sankar "**Staging of Cervical Cancer with Soft Computing**," IEEE Transactions On Biomedical Engineering, Vol. ٤٧, No. ٧, July ٢٠٠٠

[٤٠] Padhraic Smyth Usama M. Fayyad, Gregory Piatetsky-Shapiro. "**From data mining to knowledge discovery: An overview**". In *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, ١٩٩٦

[٤١] R. Santos, J.C. Nievola and A.A. Freitas. "**Extracting comprehensible rules from neural networks via genetic algorithms,"** ٢٠٠٠ IEEE Symp. on Combinations of Evolutionary Computation and Neural Networks (ECNN-٢٠٠٠), ١٣٠-١٣٩. San Antonio, TX, USA. May ٢٠٠٠.

[٤٢] R.R.F. Mendes, F.B. Voznika, A.A. Freitas and J.C. Nievola. "**Discovering fuzzy classification rules with genetic programming and co-evolution,"** Principles of Data Mining and Knowledge Discovery - Lecture Notes in Artificial Intelligence ٢١٦٨, pp. ٣١٤-٣٢٥. Springer-Verlag, ٢٠٠١.

[٤٣] R.S. Parpinelli, H.S. Lopes and A.A. Freitas. "**Data Mining with an Ant Colony Optimization Algorithm,"** IEEE Trans. on Evolutionary Computation, special issue on Ant Colony algorithms. ٢٠٠٢

[٤٤] S. K. Pal, S. Mitra, and P. Mitra, "**Rough fuzzy MLP: Modular evolution, rule generation and evaluation**," IEEE Trans. Knowledge Data Eng., ٢٠٠١.

[٤٥] S. Mitra, P. Mitra, and S. K. Pal, "**Data Mining In Soft Computing Framework: A Survey**," IEEE Transactions On Neural Networks, Vol. ١٣, No. ١, January ٢٠٠٢.

[٤٦] S. Mitra, P. Mitra, and S. K. Pal, "**Evolutionary Modular Design of Rough-Knowledge-Based Network Using Fuzzy Attributes**," Neurocomput., vol. ٣٦, pp. ٤٥–٦٦, ٢٠٠١.

[٤٧] T. Nomura; and T. Miyoshi, "**An Adaptive fuzzy rule extraction using Hybrid Model of Fuzzy Self-Organizing Map and The Genetic Algorithm With numerical Chromosomes,"** Kyoto ٦١٩-٠٢, Japan, ١٩٩٨.

[٤٨] U. Fayyad and R. Uthurusamy, "**Data mining and knowledge discovery in databases**," Commun. ACM, vol. ٣٩, pp. ٢٤–٢٧, ١٩٩٦.

[٤٩] W. Romao, A.A. Freitas and R.C.S. Pacheco. "**A Genetic Algorithm for Discovering Interesting Fuzzy Prediction Rules: applications to science and technology data,"**. Genetic and Evolutionary Computation Conf. (GECCO-٢٠٠٢). New York, July ٢٠٠٢.