Abstract

Most of the existing association rule mining algorithms are able to extract knowledge from databases with attributes of binary values. However, in real-world applications data are usually composed of continuous values such as height, length or weight. If the attributes are continuous, the algorithms commonly are integrated with a discretization method that transforms them into discrete attributes. As a result, the user most often must specify the number of intervals, or provide some heuristic rules to be used when discretization is used, then it is difficult to get the highest attribute interdependency and at the same time to get the lowest number of intervals. On the other hand, by means of discretization the so-called sharp boundary problem comes out, where the values that are close to the borders are under- or overestimated.

Recently, a novel association rule mining method using the evolutionary technique “Genetic Network Programming (GNP)” has been proposed. However, it was developed for dealing with binary values. GNP is an evolutionary optimization technique that uses directed graph structures as solutions instead of strings (Genetic Algorithms) or trees (Genetic Programming). Therefore, GNP can deal with complex problems more effectively and efficiently than GA and GP by using the higher expression ability of graph structures.

The aim of this thesis is to propose an improved association rule mining method based on Genetic Network Programming able to deal with continuous data. The improved method means the use of adaptation and self-adaptation strategies of genetic operators in GNP in order to enhance the performance of the conventional association rule extraction system. Adaptation behavior in GNP is related to adjust the setting of control parameters such as the proportion of crossover and mutation. The aim is not only to find suitable adjustments but to do it efficiently. Regarding to self-adaptation, the algorithm controls the setting of these parameters themselves embedding them into an individual’s genome and evolving them, and it usually changes the structure of the evolution which is typically static. Specifically, self-adaptation of crossover
and mutation operators in GNP aiming to change the rate of them by evolution is studied in this thesis. The results have shown that the proposed method extracts the important association rules in the database effectively. It is also found that the optimum values about genetic operators such as the proportion and rate of crossover and mutation were obtained.

To start dealing with continuous attributes, an association rule mining algorithm based on GNP with evolving thresholds is proposed. This method is able to deal with continuous values directly, that is, without using any discretization method as a preprocessing step. Additionally, the threshold values are evolved by mutation in every generation in order to obtain as many association rules as possible. Results have shown that the proposed algorithm extracts important association rules in the database effectively and can evolve both GNP and threshold parameters concurrently.

Furthermore, a fuzzy association rule mining approach that integrates GNP and fuzzy set theory is proposed. The use of fuzzy sets for describing association between data extends the types of relationships that may be represented, facilitates the interpretation of rules in linguistic terms and avoids unnatural boundaries in the partitioning of the attribute domains. Linguistic variables with linguistic terms can contribute in a substantial way to the design of association rules discovery and the analysis of data. Besides, fuzzy logic serves robust data mining models in case of false, inconsistent, and missing data. The proposed GNP-Fuzzy algorithm is based on the evolution of fuzzy association rule sets, where the parameters of the fuzzy membership functions are also evolved becoming different rule by rule. As a result, more effective and stronger rules are extracted according to the progress of the evolution. How the proposed fuzzy association rule mining can be applied to classification problems has been also studied in this thesis. The proposed fuzzy associative classifier has been compared with several relevant algorithms and the experimental results have shown the advantages and effectiveness of the proposed model.
Table of Contents

Abstract ......................................................................................... i

Chapter 1: Introduction ................................................................. 1
  1.1 Association Rule Mining ....................................................... 1
  1.2 Motivation and Objective ................................................... 2
  1.3 Structure of this thesis ....................................................... 4

Chapter 2: Genetic Network Programming (GNP)-based Data Mining Method ................................................................. 7
  2.1 Introduction ......................................................................... 7
  2.2 Chromosome Representation of GNP .................................... 7
  2.3 Genetic Operators in GNP .................................................. 9
  2.4 GNP structure for Association Rule Mining ......................... 10
  2.5 Advantages of GNP for Association Rule Mining ................. 12
  2.6 Completeness of a Data Mining algorithm ......................... 13

Chapter 3: Adaptation and Self-Adaptation Mechanisms in Genetic Network Programming for Mining Association Rules ......................... 15
  3.1 Introduction ......................................................................... 15
  3.2 Adaptation and Self-adaptation Mechanisms in GNP for Association Rule Mining ......................................................... 16
  3.3 Adaptation of Genetic Operators in GNP ............................. 17
  3.4 Self-Adaptation of Genetic Operators in GNP ..................... 18
  3.5 Simulations ......................................................................... 19
  3.6 Summary ............................................................................ 27

Chapter 4: Association Rule Mining using Genetic Network Program-
Chapter 4: Mining with Evolving Thresholds

4.1 Introduction ................................................. 29
4.2 Literature review ........................................... 30
4.3 Association Rule Mining using GNP with Evolving Thresholds .... 31
  4.3.1 Determining threshold values for the attributes .......... 31
  4.3.2 GNP structure using threshold values .................. 33
  4.3.3 Calculation of the Support, Confidence and $\chi^2$ measurements using GNP .... 35
  4.3.4 Fitness of GNP ............................................. 36
4.4 Simulations .................................................. 37
4.5 Summary ..................................................... 43

Chapter 5: Mining Fuzzy Association Rules using Genetic Network Programming ................................................................. 45

5.1 Introduction .................................................. 45
5.2 Literature review ........................................... 46
  5.2.1 Mining Fuzzy Association Rules ......................... 46
  5.2.2 Mining both Fuzzy Membership Functions and Fuzzy Association Rules .......... 47
5.3 Mining Fuzzy Association Rules using GNP ....................... 48
  5.3.1 Handling continuous attributes using Fuzzy Sets .......... 49
  5.3.2 GNP structure for mining fuzzy association rules ......... 51
  5.3.3 Updating the pool of Fuzzy Rules ....................... 54
  5.3.4 Flow of the Fuzzy Association Rule Mining ............... 55
5.4 Simulations .................................................. 57
5.5 Summary ..................................................... 58

Chapter 6: Genetic Network Programming for Extracting and Applying Fuzzy Classification Rules ................................................................. 63

6.1 Introduction .................................................. 63
6.2 Class Association Rule Mining .................................. 64
  6.2.1 Fuzzy Membership Functions for handling continuous Data .... 64
  6.2.2 GNP for Fuzzy Class Association Rule Mining ............. 66
  6.2.3 Fitness function ............................................. 69
  6.2.4 Calculation of measurements using GNP .................. 69
# List of Tables

2.1 An example of binary database. ........................................... 11

3.1 t-test results on the number of association rules stored in the pool. ... 27

4.1 Upper and lower limits of threshold ($\alpha_i=1.0$) .................................. 33
4.2 Support and confidence of the association rules. .......................... 36

5.1 Support and confidence of the fuzzy association rules. ................. 54
5.2 UCI ML datasets used in the experiments. ................................. 57
5.3 Example of the fuzzy rule updating in Class (+) of CRX DB. ....... 61

6.1 Support and confidence of the fuzzy class association rules. .......... 70
6.2 Multi-class datasets used in the experiments. .......................... 73
6.3 Parameters for GNP-based fuzzy class association rule mining. ....... 73
6.4 Classification accuracy (%) of GNP-Fuzzy algorithm and relevant association rule-based classification models. ......................... 78
6.5 Classification accuracy (%) of GNP-Fuzzy algorithm and other evolutionary classification models. .............................................. 78
6.6 Classification accuracy (%) of GNP-Fuzzy algorithm and other classification models. ............................................................... 79
6.7 Number of classification rules of GNP-Fuzzy algorithm with support value of 0.01 and other algorithms. ................................. 79
6.8 Number of classification rules of GNP-Fuzzy algorithm with different support values. ............................................................... 80
List of Figures

2.1 The basic structure of GNP individual. . . . . . . . . . . . . . . . . . . . . . . . . 8
2.2 Gene structure of GNP (node \( i \)). . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
2.3 Crossover . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
2.4 Mutation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
2.5 Basic structure of GNP for mining association rules from databases
   with binary data. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
3.1 Number of association rules in the pool in 30 simulations using con-
   ventional GNP. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
3.2 Average of number association rules in the pool over 30 simulations
   using conventional GNP. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
3.3 Average of the fitness value over 30 simulations using conventional GNP. 21
3.4 Number of association rules in Simulation 2. . . . . . . . . . . . . . . . . . . . . . . 22
3.5 Average of number association rules in Simulation 2. . . . . . . . . . . . . . . . . 22
3.6 Average of the fitness value over 30 simulations in Simulation 2. . . . . . . 22
3.7 Proportion of each genetic operator executed in Simulation 2. . . . . . . . . . 22
3.8 Number of association rules in Simulation 3. . . . . . . . . . . . . . . . . . . . . . 24
3.9 Average number of association rules in Simulation 3. . . . . . . . . . . . . . . . . 24
3.10 Average fitness value in Simulation 3. . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
3.11 Proportion of each genetic operator executed in Simulation 3. . . . . . . . . . 24
3.12 Number of association rules in Simulation 4. . . . . . . . . . . . . . . . . . . . . . . 25
3.13 Average number of association in Simulation 4. . . . . . . . . . . . . . . . . . . . . 25
3.14 Average of fitness value in Simulation 4. . . . . . . . . . . . . . . . . . . . . . . . . . . 25
3.15 Proportion of each genetic operator executed in Simulation 4. . . . . . . . . . 25
3.16 Change of the rate of Crossover in self-adaptation mechanism. . . . . . . . . . 26
3.17 Change of the rate of Mutation-1 in self-adaptation mechanism. . . . . . . . . 26
3.18 Change of the rate of Mutation-2 in self-adaptation mechanism. . . . . . . . . 26
6.4 Deterministic transition from one judgment node to another.

6.5 Deterministic transition from one judgment node to another comparing the fuzzy value with 0.5.

6.6 The connection of the judgment nodes is changed by mutation rate of $P_{m1}$.

6.7 The function of the judgment nodes is changed by mutation rate $P_{m2}$.

6.8 Flowchart of the proposed fuzzy class association rule mining.

6.9 Number of rules in Waveform.

6.10 Number of rules in CRX.

6.11 Number of rules in Ionosphere.

6.12 Number of rules in Heart.

6.13 Number of rules in Pima.

6.14 Number of rules in Pageblocks.

6.15 Number of rules in Glass.

6.16 Number of rules in Iris.

6.17 Accuracy in Waveform.

6.18 Accuracy in CRX.

6.19 Accuracy in Ionosphere.

6.20 Accuracy in Heart.

6.21 Accuracy in Pima.

6.22 Accuracy in Pageblocks.

6.23 Accuracy in Glass.

6.24 Accuracy in Iris.

6.25 Evolution of the fuzzy parameters by uniform and non-uniform mutation.

6.26 Classification accuracy for CRX DB for different minimum support values.
Chapter 1

Introduction

1.1 Association Rule Mining

The evolution of database systems and information technology facilitated companies to record huge amounts of data from their customers. The abundance of data, coupled with the need for powerful data analysis tools, has been described as a *data-rich but information-poor* situation [1]. Therefore, *knowledge discovery in databases (KDD)*, defined as the nontrivial extraction of implicit, previously unknown and potentially useful information from data [2], has become a process of considerable interest in recent years. The KDD process generally consists of three phases: preprocessing, data mining and post-processing [3]. Among them, data mining plays a decisive role in KDD. Data mining means the discovery of interesting information, patterns or trends from a large database or datawarehouse. Since 1990’s many data mining techniques have been developed that go beyond simple analysis, contributing greatly to business strategies, knowledge bases and scientific research. Therefore, data mining has become a central topic in databases and is referred as a multi-disciplinary field involving machine learning, statistics, databases, artificial intelligence, information retrieval and visualization. Although data mining is considered sometimes as synonym of knowledge discovery, it is more accurate to regard data mining as a particular task within the general knowledge discovery process.

Since the introduction by Agrawal et al. in 1993 [4], association rules mining has received great interest by the data mining community. Association rules show attributes that occur frequently together in a given dataset. A typical and widely used example of association rule mining is market basket analysis, where the goal is to mine
patterns describing the customer’s purchase behavior. Generally, an association rule is an expression \((X \Rightarrow Y)\), where \(X\) (antecedent) is a set of items and \(Y\) (consequent) is usually a single item, but could be a set of items as well. Support and confidence are the two most important quality measures for evaluating the interestingness of rules. An association rule could be "If a customer buys butter, he will mostly buy milk as well" ("Butter \(\Rightarrow\) Milk") \([support = 0.2, confidence = 0.8]\). In addition to the above example, association rule mining has attracted many research efforts along with a large number of applications in various fields such as finance, stock market, medicine, manufacturing, e-business, intrusion detection, bioinformatics, etc.

Last years, several techniques have been applied to association rule mining, such as expert systems, artificial neural networks, linear programming, database systems, and evolutionary computing. Among these approaches, evolutionary computing (EC) has emerged as a promising technique for extracting comprehensible association rules in data mining. EC is a computational technique inspired by the natural evolution process that imitates the mechanism of natural selection and survival of the fittest. The motivation for applying evolutionary algorithms to data mining lays in they are robust search methods which perform a global search in the space of candidate solutions. In contrast, most association rule induction methods perform a local, greedy search in the space of candidate rules. Intuitively, the global search of evolutionary algorithms can discover interesting rules and patterns that would be missed by the greedy search performed by most rule induction methods [5].

1.2 Motivation and Objective

Extensive research has been carried out in order to develop enhanced association rule mining methods. However, current algorithms still have some limitations such as:

- **Handling continuous data**: A problem of classical association rule mining algorithms is that they are mostly limited to handle attributes with binary values. Thus, rules can only be mined from binary databases, where an item either exists in a transaction or it does not exist. However, in most real applications, databases are usually composed of continuous attributes such as income, age, quantity of products and so on. This fact guided to the use of discretization methods, where the continuous attributes are split into intervals and the item becomes either the member or non-member of those intervals. This leads to the
so-called sharp boundary problem, where the values that are close to the borders are under- or overestimated. Alternative ways to deal with continuous values are the introduction of decision trees. Nevertheless, their instability, i.e., small data deviations may require a significant reconstruction of the decision tree and in cases of large data tables, they are often inefficient in both computation and description aspects.

- **Limitations of the support-confidence framework**: Often, many interesting rules can be found using support-confidence thresholds. However, recent studies show that the support and confidence measures are insufficient for filtering out uninteresting association rules, for instance, even strong association rules can be uninteresting and misleading.

- **Handling a large set of discovered rules**: The common property of conventional algorithms is that they are looking for frequent item sets in a first phase and generate rules in a second one, instead of generating a large number of association rules. On the other hand, association rule mining methods based on evolutionary algorithms require the encoding scheme of the individuals for representing rule sets. That is, each individual represents the association rule encoding the rule itself into the individuals. Therefore, the so-called bloating problem (the uncontrolled growth of the average size of an individual in the population) is usually found for discovering a large set of rules.

In the past, a novel association rule mining method using the evolutionary technique “Genetic Network Programming (GNP)” has been proposed [6, 7]. However, it was developed for dealing with binary values only. GNP is graph based evolutionary algorithm derived from Genetic Algorithms (GA) and Genetic Programming (GP). Thus, the main difference among them is the representation of the solutions. GA evolve strings as solutions and it is mainly applied to optimization problems. GP was formulated later in order to expand the expression ability of GA by using tree structures. GNP uses directed graph structures as solutions, therefore GNP can deal with complex problems more effectively and efficiently than GA and GP [8, 9].

The objective of this thesis is to propose an improved, robust and unique association rule mining method based on Genetic Network Programming capable of overcome the above problems, principally dealing with databases with continuous data. The improved method means the use of adaptation and self-adaptation strategies of
genetic operators in GNP in order to enhance the performance of the conventional association rule extraction system. Adaptation and self-adaptation behavior in GNP is related to adjust dynamically the setting of control parameters such as the proportion of crossover and mutation. The aim is not only to find suitable adjustments but to do it efficiently. Regarding to continuous data, two approaches are proposed: (1) An association rule mining algorithm based on GNP with evolving thresholds. This method is able to deal with continuous values directly, that is, without using any discretization method as a preprocessing step. (2) A fuzzy association rule mining approach that integrates GNP and fuzzy set theory. The proposed GNP-Fuzzy algorithm is based on the evolution of fuzzy association rule sets, where the parameters of the fuzzy membership functions are also evolved becoming different rule by rule. How the proposed fuzzy mining technique can be applied to classification problems is also studied in this thesis.

1.3 Structure of this thesis

The structure of this thesis is summarized as follows:

- Chapter 2 describes the conventional association rule mining method based on Genetic Network Programming. A general description, characteristics, advantages of GNP and the reasons of selecting a GNP-based mining method are explained.

- Chapter 3 discusses how adaptation and self-adaptation strategies are applied to the crossover and mutation process in order to increase the number of interesting association rules stored in the pool. In other words, the conventional GNP-based association rule mining algorithm is extended in terms that (1) an adaptation mechanism of GNP in order to change automatically the proportion of the crossover and mutation of the individuals and (2) self-adaptation of the crossover and mutation rates, which are typically static, in order to improve the performance of association rule extraction of GNP.

- Chapter 4 introduces the proposed association rule mining algorithm based on GNP with evolving threshold values, where attributes in databases correspond to judgment nodes in GNP and each continuous attribute is checked whether its value is greater than a threshold value. Moreover, the threshold values are
evolved by mutation in order to obtain as many association rules as possible. The simulation results have shown that the proposed method extracts important association rules in the database, effectively.

- Chapter 5 describes the proposed association rule mining approach that integrates Genetic Network Programming and fuzzy set theory for mining interesting fuzzy rules from given continuous data. Therefore, the proposed GNP-Fuzzy algorithm is based on the evolution of fuzzy association rule sets, where the parameters of the fuzzy membership functions are also evolved becoming different rule by rule. As a result, more effective and stronger rules are extracted according to the progress of the evolution.

- Chapter 6 describes how to extract fuzzy class association rules and how to classify the data. The proposed algorithm consists of two major phases: (1) generating fuzzy class association rules using GNP (2) building a classifier based on the extracted fuzzy rules. In the first phase, the main task is to extract fuzzy class association rules using a GNP-based algorithm. In the second phase, all of the generated fuzzy rules in the pool are used to predict the class of the test set. The performance of the proposed algorithm has been evaluated by selecting eight public-domain datasets from UCI (University of California at Irvine). The experimental results have shown the advantages and effectiveness of the proposed model.

At last, some conclusions are given for describing the main achievements of this thesis. Generally speaking, the simulation results have shown that the proposed algorithms extract important association rules in the database effectively and are capable of evolving GNP and thresholds/fuzzy parameters concurrently. Regarding to the GNP-based classifier, the achieved accuracy reveals the effectiveness of the proposed fuzzy class association rule mining and classification.
2.1 Introduction

Genetic Network Programming (GNP) is one of the evolutionary algorithms inspired by biological evolution. Thus, GNP is an extension of Genetic Algorithms (GA) and Genetic Programming (GP). The main difference between them is the representation of the solutions. GA evolve strings as solutions and it is mainly applied to optimization problems. GP was formulated later in order to expand the expression ability of GA by using tree structures. GNP uses directed graph structures as solutions instead of strings or trees, therefore GNP can deal with complex problems more effectively and efficiently than GA and GP [8, 9]. The aim of developing GNP is to deal with dynamic environments efficiently by using the higher expression ability of graph structures. GNP produced many novel and outstanding results in the areas such as Data Mining [6, 7], Intelligent Mobile Robot [8], Multi Agent System [10], Elevator Group Supervisory Control System [11], Stock Market Prediction [12], Automatic Program Generation [13], and so on.

2.2 Chromosome Representation of GNP

The basic structure of GNP is shown in Figure 2.1. GNP is composed of three types of nodes that are connected on a network structure: a start node, judgment nodes (diamonds), and processing nodes (circles). Judgment nodes are the set of $J_1$, $J_2$, …,
Chapter 2. Genetic Network Programming (GNP)-based Data Mining Method

\( J_p \), which work as if-then conditional decision functions and they return judgment results for assigned inputs and determine the next node to be executed. Processing nodes are the set of \( P_1, P_2, \ldots, P_q \), which work as action/processing functions. The start node determines the first node to be executed. The nodes transition begins from the start node, however there are no terminal nodes. After the start node is executed, the next node is determined according to the node’s connections and judgment results.

![Figure 2.1: The basic structure of GNP individual.](image)

The gene structure of GNP (node \( i \)) is shown in Figure 2.2. \( NT_i \) describes the node type, \( NT_i = 0 \) when node \( i \) is start node, \( NT_i = 1 \) when node \( i \) is judgment node and \( NT_i = 2 \) when node \( i \) is processing node. \( ID_i \) is an identification number, for example, \( NT_i = 1 \) and \( ID_i = 1 \) mean node \( i \) is \( J_1 \). \( C_{i1}, C_{i2}, \ldots \), denote the nodes which are connected from node \( i \) firstly, secondly, \ldots, and so on depending on the arguments of node \( i \). \( d_i \) and \( d_{ij} \) are the delay time. They are the time required to execute the judgment or processing of node \( i \) and delay time from node \( i \) to node \( C_{ij} \), respectively.

![Figure 2.2: Gene structure of GNP (node \( i \)).](image)
2.3 Genetic Operators in GNP

The main genetic operators executed in GNP individuals are crossover and mutation.

- Crossover: is the genetic operation that generates two new offspring from two parents by exchanging the genetic information [see Figure 2.3]. The selected nodes and its connections in two parents are swapped each other by crossover rate $P_c$. One point crossover, several points crossover and uniform crossover are established in GNP.

- Mutation: is executed in one GNP individual in order to generate a new one. There are two types of mutation in GNP [see Figure 2.4]: mutation of the connection, where a node branch is selected with a probability of $P_{m1}$ and reconnected to another node, and mutation of the content, where a node function is selected with a probability of $P_{m2}$ and changed to another one.

![Figure 2.3: Crossover](image)

The characteristics of GNP are described as follows.

- The judgment and processing nodes are repeatedly used in GNP, therefore the structure becomes compact and an efficient evolution of GNP is obtained.

- Since the number of nodes are defined in advance, GNP can find the solutions of the problems without bloating, which can be sometimes found in Genetic Programming (GP).
Chapter 2. Genetic Network Programming (GNP)-based Data Mining Method

Figure 2.4: Mutation

- Nodes that are not used at the current program executions will be used for future evolution.

- GNP is able to cope with partially observable Markov processes.

- The node transition in GNP individual is executed according to its node connections without any terminal nodes, thus the history of the node transition affects the next node to be used.

The following issues have to be determined in a GNP system: (1) the number of judgment and processing nodes in the generated GNP individuals, therefore all the programs in a population have the same number of nodes, (2) the number of branches per judgment node, (3) fitness function for evaluating GNP individuals, (4) generation of the initial population, (5) selection strategies of the population of the next generation, (6) how crossover and mutation operations are carried out and how often these operations are performed, (7) condition for terminating the evolution process.

2.4 GNP structure for Association Rule Mining

A method for extracting association rules using Genetic Network Programming was already proposed [6, 7]. However, it was developed for dealing with binary values. The
basic structure of GNP for mining association rules from databases with binary data is shown in Figure 2.5. GNP examines the attribute values of database tuples using judgment nodes and calculates the measurements of association rules using processing nodes. Attributes and their values correspond to judgment nodes and their judgments in GNP, respectively. Therefore, the connections of judgment nodes are represented as association rules. GNP’s structure allows to calculate the support, confidence, $\chi^2$ and many other measurements easily in order to quantify the significance of the rules to be integrated into the model. Judgment node determines the next node by a judgment result (Yes/No). In Figure 2.5, $P_1$ is a processing node and is a starting point of association rules, each processing node has an inherent numeric order ($P_1$, $P_2$, . . ., $P_s$) and is connected to a judgment node. Yes-side of the judgment node is connected to another judgment node. Judgment nodes can be reused and shared with some other association rules because of GNP’s characteristics. No-side of the judgment node is connected to the next numbered processing node. The above explanation is illustrated using the following example. In Table 2.1, the tuple 1 $\in$ TID satisfies $A = 1$ and $B \neq 1$, therefore the node transition from $P_1$ to $P_2$ occurs in Figure 2.5. If the examination of the connection from the starting point $P_s$ ends, then GNP examines the tuple 2 $\in$ TID. Thus, all tuples in database will be examined. The total number of tuples moving to Yes-side at each judgment node is calculated for every processing node, which is a starting point for calculating association rules. All GNP individuals are searched in parallel at the same time. If Yes-side connection of judgment nodes continues and the number of the judgment nodes becomes a cutoff value (maximum number of attributes in extracted association rules), then Yes-side connection is transferred to the next processing node obligatorily. In Figure 2.5, N is the number of total tuples, and $a$, $b$, $c$ and $d$ are the number of tuples moving to Yes-side at each judgment node. These counts are used for calculating measurements such us support and confidence of each association rule.

<table>
<thead>
<tr>
<th>TID</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Chapter 2. Genetic Network Programming (GNP)-based Data Mining Method

Figure 2.5: Basic structure of GNP for mining association rules from databases with binary data.

The following genetic operators are executed to GNP individuals:

- **Crossover**: Operator producing offspring from parents. Uniform crossover is used. Judgment nodes are selected as crossover nodes with the probability of $P_c$. Two parents exchange the gene of the corresponding crossover nodes.

- **Mutation-1**: Operator that affects one individual. The connection of the judgment nodes is changed by mutation rate of $P_{m1}$.

- **Mutation-2**: Operator that also affects one individual. The function of the judgment nodes is changed by mutation rate $P_{m2}$.

All programs in a population have the same number of nodes, but the node with the same node number need not have the same function. All the connections of the processing nodes are changed randomly in order to extract rules efficiently.

2.5 Advantages of GNP for Association Rule Mining

In this thesis, the conventional GNP-based data mining method is extended and improved in order to mine interesting association rules from given continuous data. The reasons of selecting a GNP-based association rule mining method are described as follows:

- There is no need to encode the association rule into the genome of an individual in GNP. Thus, GNP-based association rule mining method is a tool for extracting a large number of rules into the general pool. Whereas, other evolutionary
techniques requires the encoding scheme of the individuals for representing rule sets.

- GNP is able to extract a large number of rules without the bloating problem found in Genetic Programming. Because of GNP’s features described above, the judgment nodes could be reused and shared with some other association rules.

- GNP, through judgment nodes, is capable of dealing with many kinds of data, not only binary data but also continuous, fuzzy, time series data and so on.

- The measurements of the rules, such as the support, confidence, $\chi^2$ and many other correlation measures can be are easily calculated by GNP individuals.

- The genetic operators executed in GNP such as crossover, mutation of the connections and mutation of the content, allows the GNP-based association rule mining method to find new rules according to the progress of the evolution. In other words, due to the genetic operators to GNP individuals, new attributes are evaluated and new relationships between attributes are found in each generation, as a result, the new rules that satisfy the minimum thresholds for the significance measures are mined.

- According to the supervisor’s requirements, GNP is able to mine an adequate number of rules through generations and store them in the general pool, this is because the rules extraction is done without identifying frequent itemsets as most Apriori-based data mining algorithms, therefore, the GNP-based association rule mining method extracts important rules for the user’s purpose in a short time.

### 2.6 Completeness of a Data Mining algorithm

While conventional association rule mining algorithms were concentrated on extracting all the rules from a given database, current models are predominantly focused on extracting a large number of rules using high quality measurements.

According to Han et al. [14], two features are considered regarding to the completeness of a data mining algorithm: (1) Generating all rules from the database. (2) Provide constraints and interestingness measurements in order to extract rules with
high quality. Then, the second feature is most important than the first one because it is not relevant to extract all the rules if most of them are uninteresting, misleading or redundant. Association rule mining is an example where the use of constraints and interestingness measurements can complement the completeness of mining by extracting association rules with high quality [14].

The conventional GNP-based data mining method extracts significant number of rules, in a short period of time. However, the final pool of rules does not contain all the association rules.

In this thesis, along with four proposed methods, the first feature was improved, respect to the conventional GNP, by proposing adaptation, self-adaptation, evolution of the thresholds and fuzzy parameters in order to increase the number of rules extracted.

Regarding to the second feature, $\chi^2$ was used in addition to the support and confidence framework. The quality of the extracted rules was proved in the proposed associative rule-based classifier because the highest accuracy was obtained in comparison with relevant methods.

Therefore, extracting a large set of association rules with high quality is one of the main objectives in this thesis.
Chapter 3

Adaptation and Self-Adaptation Mechanisms in Genetic Network Programming for Mining Association Rules

3.1 Introduction

Evolutionary algorithms operate on the basis of the populations of individuals. Their performance depends on the characteristics of the population’s distribution. Based on the mechanism of parameter’s adaptation, three subclasses are distinguished: deterministic, adaptive, and finally self-adaptive algorithms [15]. A deterministic adaptation is used if the strategy parameter is changed according to a deterministic rule without taking into account any current information in the evolutionary algorithm itself. Examples of this class of adaptation are the time-dependent change of the mutation rates proposed by Holland [16] and the cooling schedule in simulated annealing.

Algorithms with an adaptive rule take the feedback from the evolutionary algorithm itself into account and change the strategy parameters accordingly. A well known member of this class is Rechenberg’s 1/5th-rule [17]. It relies on counting the successful and unsuccessful mutations for a certain number of generations. If more than 1/5 mutations leads to an improvement the mutation strength is increased and decreased otherwise. The aim was to stay in the so-called evolution window
guaranteeing nearly optimal progress.

Self-Adaptation aims at biasing the distribution towards appropriate regions of the search space maintaining sufficient diversity among individuals in order to enable further evolvability. Generally, this is achieved by adjusting the setting of strategy parameters. Strategy parameters can be of various forms ranging from crossover and mutation rates, recombination probabilities, and population size to selection operators. The basic idea of explicit self-adaptation consists in incorporating the strategy parameters into the individual’s genome and evolving them along with the parameters. Crossover and mutation are traditionally regarded as the main diversification mechanisms in evolutionary algorithm and most efforts to self-adaptation stem from this area.

3.2 Adaptation and Self-adaptation Mechanisms in GNP for Association Rule Mining

In this chapter, how adaptation and self-adaptation strategies can be put into a GNP context is studied. Consequently, the conventional GNP-based data mining mining algorithm, described in section 2.3, is extended and two strategies are proposed (1) an adaptation mechanism in order to change automatically the number of GNP individuals for crossover and mutation genetic operators and (2) self-adaptation of the crossover and mutation rates, which are typically static, in order to improve the performance of association rule extraction of the GNP. The adaptation and self-adaptation strategies considered in this chapter are applied to crossover and mutation process and change the proportions of GNP individuals for crossover, mutation-1 and mutation-2 in each generation in order to increase the number of interesting association rules stored in the pool.

Therefore, the most important characteristics of this work are described as follows:

- Determining which genetic operators (crossover, mutation-1 or mutation-2) contribute more to the evolution of GNP individuals in each generation and which of them is most likely to find new rules.

- An exponential smoothing method is used for obtaining the proportion of individuals needed for each genetic operator in each generation.
• Determining the optimum rates of crossover, mutation-1 and mutation-2 for improving the performance of the data mining using GNP.

• In self-adaptation, crossover and mutation rates are mutated according to a uniform probability distribution over a set of possible values.

• Introducing adaptation and self-adaptation mechanisms for improving the general performance of the GNP based data mining method, especially increasing the number of rules extracted and decreasing the number of generations where the extraction starts.

3.3 Adaptation of Genetic Operators in GNP

In the conventional GNP-based data mining method, the proportions of GNP individuals for each genetic operator at each generation remain static. In contrast, the adaptive strategy applied in this chapter, changes dynamically the number of GNP individuals to be produced by each genetic operator according to their contribution to the number of association rules extracted. In order to obtain the new proportions at each generation, a GNP individual is complemented with a bit string to code the genetic operator used to create the individual, such as: crossover, mutation-1 or mutation-2. Thus, GNP individuals are distinguished each other due to its inherent genetic operators. These operators are executed to the gene of judgment nodes and processing nodes of GNP. At an earlier stage when no rules are extracted, the individuals are ranked by the fitness, then upper 1/3 individuals are selected and they are reproduced three times, as a result, three kind of genetic operators like crossover, mutation-1 and mutation-2 are executed to them at the same proportion. In other words, before finding association rules, each genetic operator is executed to the same number of individuals, however as the generation goes on, the number of individuals obtained by each genetic operation is calculated based on its contribution to the rules extracted in the pool.

The proportion for Crossover, Mutation-1 and Mutation-2 is calculated as follows:

\[ P_{c(t)} = P_{c(t-1)} + \alpha (Rc - P_{c(t-1)}) \], \quad (3.1) 
\[ P_{m1(t)} = P_{m1(t-1)} + \alpha (R_{m1} - P_{m1(t-1)}) \], \quad (3.2) 
\[ P_{m2(t)} = P_{m2(t-1)} + \alpha (R_{m2} - P_{m2(t-1)}) \]. \quad (3.3)
Chapter 3. Adaptation and Self-Adaptation Mechanisms in Genetic Network Programming for Mining Association Rules

Where, $P_c(t)$, $P_{m1}(t)$, $P_{m2}(t)$: Proportion of individuals to be produced by crossover, mutation-1 and mutation-2, respectively in generation $t$. $\alpha$: Smoothing coefficient. $R_c$, $R_{m1}$, $R_{m2}$: number of association rules in the pool, produced by crossover, mutation-1 and mutation-2, respectively.

The above is illustrated generating 120 GNP individuals for evolution. At first, the individuals are ranked by their fitness value, then, the top 40 individuals are selected and 120 individuals are reproduced using crossover, mutation-1 and mutation-2, which have their own proportion to reproduce individuals. That is, depending on the number of individuals calculated by $P_c(t)$, $P_{m1}(t)$, $P_{m2}(t)$, new offspring is produced by crossover, mutation-1 and mutation-2, respectively.

3.4 Self-Adaptation of Genetic Operators in GNP

A traditional evolutionary computation scheme uses only a limited, fixed probability of reproductive operators, such as mutation and crossover. This basic paradigm remains unchanged in the majority of studies that treat parameters of reproductive operators as static and non-evolvable entities [18]. In contrast, the key idea of a self-adaptive approach is to consider a search for a best evolutionary algorithm as an optimization problem to be solved by evolutionary computation. In a globalistic interpretation of this idea [19], a set of base-level evolutionary algorithms, each with its own set of parameters and operators are treated as the evolving population of individuals at the meta-level in a dynamic and evolvable environment. In other words, this alternative paradigm treats reproductive operators on their own as individuals subject to evolution.

In this section, the evolution of genetic operators in GNP is introduced in order to find the optimal rates for crossover and mutation. Thus, self-adaptation of crossover and mutation rates in GNP is carried out by introducing an additional parameter which controls the rate of these genetic operators. More concretely, the rates of the genetic operators, that is, the Crossover rate ($r_c$), Mutation-1 rate ($r_{m1}$) and Mutation-2 rate ($r_{m2}$) of each individual, are mutated in every generation. Initially, crossover and mutation rates are fixed to a value of $r_c = 15/78$, $r_{m1} = 1/3$ and $r_{m2} = 1/5$, then, the three parameters are mutated according to a uniform probability
distribution over a set of a possible values. These possible values were determined for every genetic operator independently. The parameter space for crossover rate was determined after experimental simulations and a possible value for crossover rate is defined as a ratio of an integer number (10, 15, 20, 25 or 30) over the total number of judgment nodes of the GNP (78) because they show better efficiency in the number of rules extracted by each individual than other values. The parameter space for Mutation-1 and Mutation-2 are the same. They were also determined by experimental simulations and the selected values were: 1/2 (high probability to carry out mutation), 1/3 and 1/5 (medium probability to carry out mutation) and 1/10 (low probability to carry out mutation). Then, the parameter space for Crossover, Mutation-1 and Mutation-2 rate is determined as follows:

1) Crossover rate \( r_c \) \( \in \{10/78, 15/78, 20/78, 25/78, 30/78\} \).
2) Mutation-1 rate \( r_{m1} \) \( \in \{1/2, 1/3, 1/5, 1/10\} \),
3) Mutation-2 rate \( r_{m2} \) \( \in \{1/2, 1/3, 1/5, 1/10\} \).

### 3.5 Simulations

To confirm the performance improvement of the GNP-based data mining with adaptation and self-adaptation mechanisms compared to the conventional method, four simulations were carried out.

The experiments were run on synthetic database including 52 attributes (\( A_j, j = 1, 2, \ldots, 52 \)). The number of tuples are 200, \( \text{support}(A_j = 1) = 0.7 \) (\( j = 1, 2, \ldots, 5 \)), and \( \text{support}(A_j = 1) = 0.5 \) (\( j = 6, 7, \ldots, 52 \)). For GNP with self-adaptation of genetic operators, the rate of mutating crossover and mutation rates is \( P = 1/3 \). The initial crossover and mutation rates are fixed at values mentioned above. The number of changing the connections of the processing nodes at each generation is 5. The algorithm was executed for 1000 generations for all the experiments studied here. The simulations are executed in the following four cases:

- **Simulation 1** demonstrates the performance of the conventional GNP-based data mining method.
- **Simulation 2** shows the performance of the GNP-based method with adaptive mechanism using the information on the number of association rules produced
by each operator in each generation.

- Simulation 3 shows the performance of the proposed method with adaptive mechanism using the information on the number of association rules produced by each operator in the latest 10 generations.

- Simulation 4 demonstrates the performance of the proposed method with adaptation mechanism using the conditions of Simulation 3 and self-adaptation mechanism using evolving crossover and mutation rates in every generation.

Simulation 1 is carried out in order to compare the performance of the proposed methods with the conventional GNP-based method. Thus, the following three figures show the performance of the conventional one. Figure 3.1, Figure 3.2 and Figure 3.3 show the number of important association rules in 30 independent trials stored in the pool, the average number of association rules stored in the pool and the average fitness value, respectively. In this simulation, the proportion of individuals for genetic operators remains the same for all generations and its results are used as a reference to compare with next simulations.

The performance of the GNP-based method with adaptive mechanism using the information on the number of association rules produced by each operator in each generation is studied in simulation 2. Figure 3.4 and Figure 3.5 show the number of important association rules in 30 independent trials stored in the pool and its average, while Figure 3.6 shows the average fitness value. On the other hand, Figure 3.7 shows the average proportion of each genetic operator. These figures show that the method with adaptation mechanism extracts the important association rules in the database more effective because it begins to extract rules in earlier generations than Simulation 1, however the number of interesting rules stored in the pool did not increase very much, almost the same as the conventional GNP. In this simulation, in earlier generations the association rules’s extraction is stronger compared to the later generations, because the proportions of genetic operators vary according to the formula of exponential smoothing. In later generations there are no new rules to extract, then the proportions of genetic operators almost remain the same. This is reflected in the total amount of association rules extracted that does not vary so much with respect to the Simulation 1.
3.5. Simulations

Figure 3.1: Number of association rules in the pool in 30 simulations using conventional GNP.

Figure 3.2: Average of number association rules in the pool over 30 simulations using conventional GNP.

Figure 3.3: Average of the fitness value over 30 simulations using conventional GNP.
Chapter 3. Adaptation and Self-Adaptation Mechanisms in Genetic Network Programming for Mining Association Rules

Figure 3.4: Number of association rules in Simulation 2.

Figure 3.5: Average of number association rules in Simulation 2.

Figure 3.6: Average of the fitness value over 30 simulations in Simulation 2.

Figure 3.7: Proportion of each genetic operator executed in Simulation 2.
Simulation 3 shows the performance of the adaptation mechanism using the information on the number of association rules produced by each operator in the latest ten generations. The simulations were executed over 30 independent trials. Figure 3.8 shows the number of important association rules stored in the pool, while Figure 3.9 shows its average. Figure 3.10 shows the average fitness value. On the other hand, Figure 3.11 shows the proportions of each genetic operator. It can be seen that mutation-2 contributes more. These figures show that the generation where the extraction of rules begins is faster than Simulation 1, the number of interesting rules obtained in the pool is increased and the average fitness is also increased compared to the conventional method. They also show that using the information of the latest ten generations leads to the fact that the genetic operator mostly used is Mutation-2 and it is used to contribute to the extraction of association rules.

Simulation 4 shows the performance of the proposed method with adaptation and self-adaptation mechanisms using the information on the number of association rules produced by each operator in the latest ten generations and changing the crossover and mutation rates. Figure 3.12, Figure 3.13, Figure 3.14 and Figure 3.15 show the number of important association rules stored in the pool, its average, the average fitness value and the proportion of each genetic operator executed over 30 independent trials, respectively. It can be seen that mutation-2 contributes more to the rule’s extraction, same as simulation 3. Thus, the proposed method using adaptation and self-adaptation is more efficient compared to the conventional one because extraction of association rules starts much earlier; the number of interesting rules obtained in the pool is increased and the average fitness is also increased compared to GNP based data mining method with only adaptation mechanism. The most important characteristics of the results of self-adaptation mechanism are: crossover and mutation rates of each individual are calculated in every generation. The rate of crossover increased a bit compared with the rate used in simulation 1. For instance, in simulation 1 the rate of crossover is \(\frac{15}{78}=0.19\), whereas in simulation 4 the average value is 0.22, as is shown in Figure 3.16. The rate of mutation-1 decreased compared with the rate used in simulation 1. That is, although, in simulation 1 the rate of mutation-1 is \(\frac{1}{3}=0.33\), the average value becomes 0.25 in simulation 4, as is shown in Figure 3.17. The rate of mutation-2 increased compared with the rate used in simulation 1. For instance, in simulation 1 the rate of mutation-2 is \(\frac{1}{5}=0.20\), while is 0.25 in simulation 4, as is shown in Figure 3.18.
Chapter 3. Adaptation and Self-Adaptation Mechanisms in Genetic Network Programming for Mining Association Rules

Figure 3.8: Number of association rules in Simulation 3.

Figure 3.9: Average number of association rules in Simulation 3.

Figure 3.10: Average fitness value in Simulation 3.

Figure 3.11: Proportion of each genetic operator executed in Simulation 3.
3.5. Simulations

Figure 3.12: Number of association rules in Simulation 4.

Figure 3.13: Average number of association in Simulation 4.

Figure 3.14: Average of fitness value in Simulation 4.

Figure 3.15: Proportion of each genetic operator executed in Simulation 4.
In order to verify the results with statistical measures, the proposed methods with adaptation and adaptation and self-adaptation mechanism are evaluated by running 30 independent trials and averaging data. Table 3.1 shows the average number of association rules obtained in the pool, its standard deviation and the results of the $t$–test between conventional GNP-based method, GNP with adaptation mechanism
and GNP with adaptation and self-adaptation mechanism.

Table 3.1: t-test results on the number of association rules stored in the pool.

<table>
<thead>
<tr>
<th>Method</th>
<th>Average</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GNP Convent. Method</td>
<td>7860.45</td>
<td>2136.57</td>
</tr>
<tr>
<td>GNP Adapt. Mech.</td>
<td>10679.20</td>
<td>2868.58</td>
</tr>
<tr>
<td>GNP Adapt. and Self-Adapt. Mech.</td>
<td>14919.20</td>
<td>3386.63</td>
</tr>
</tbody>
</table>

From Table 3.1, it can be seen that the t-test is small enough to indicate a significant difference between the proposed GNP-based method with adaptation and self-adaptation strategies and the conventional one. Therefore, it is clarified that proposed methods outperforms the conventional method based on GNP.

3.6 Summary

In this chapter, a method of association rule mining using Genetic Network Programming with adaptation and self-adaptation mechanism was developed. The adaptation and self-adaptation strategies are applied to crossover and mutation process and evolve the rates and proportions of GNP individuals for crossover, mutation-1 and mutation-2 in each generation in order to increase the number of interesting association rules stored in the pool. The results show that the proposed methods extracts the important association rules in the database effectively. Moreover, it is also found that optimum values for the genetic operators such as rates and the proportion of crossover and mutation could be obtained. It is also shown that the overall performance of the proposed system is better than the conventional method based on GNP.

Therefore, the main contributions of this work are described as follows:

- Determining which genetic operators contribute more to the evolution of individuals and which of them is most likely to find new rules.
• Determining mutation and crossover rates constitutes an optimization problem by itself. In this thesis, self-adaptation is introduced in order to obtain the optimum rates of crossover, mutation-1 and mutation-2 for improving the performance of the conventional data mining method using GNP. The crossover and mutation rates are mutated according to a uniform probability distribution over a set of possible values.

• The proposed adaptation and self-adaptation mechanisms extracts the important association rules in the database more effective because it begins to extract rules in earlier generations than the conventional one. In other words, in earlier generations the association rules’s extraction is stronger compared to the later generations, because the proportions of genetic operators vary according to the formula of exponential smoothing. In later generations there are no new rules to extract, then the proportions of genetic operators almost remain the same.

• It was found that mutation-2 contributes more to the rule’s extraction. Thus, the proposed methods are more efficient because extraction of association rules starts much earlier, the number of interesting rules obtained in the pool is increased and the average fitness is also increased compared to the conventional method.

• Adaptation and self-adaptation mechanisms improves the general performance of the conventional GNP-based data mining method, especially increasing the number of rules extracted and decreasing the number of generations where the extraction starts. Concretely speaking, the adaptation and self-adaptation mechanism adapt automatically the rates and proportion of GNP individuals for crossover, mutation-1 and mutation-2, determining which genetic operator contribute more to the evolution of individuals and which of them is most likely to find new rules.
Chapter 4

Association Rule Mining using Genetic Network Programming with Evolving Thresholds

4.1 Introduction

Data Mining is the process of extracting useful hidden knowledge from large volumes of data and its results can be used in decision support systems. Several data mining algorithms have been recently developed; one example is the association rule mining, which discovers associations among items encoded within a database. The Apriori algorithm [20] is the most famous example of association rule mining algorithms. Using this model, candidate patterns which receive sufficient support (occur sufficiently often) from the database are considered for extracting rules. A restriction of this and many other association rules mining algorithms is that they are limited to handle the attributes with binary values. However, in most real life applications, databases contain many other attribute values besides 0 and 1, since data are usually composed of continuous values such as height, length or weight. If the attributes are continuous, the algorithms are commonly integrated with a discretization method as a preprocessing step that transforms them into discrete attributes. Conversely, by using these kind of methods, the user most often must specify the number of intervals, or provide some heuristic rule to be used, then it is difficult to get the highest attribute interdependency and at the same time to get the lowest number of intervals. On the other hand, the sharp boundary problem is found when discretization is used because the
values that are close to the borders might be over or under estimated.

In this chapter, an association rule mining algorithm based on GNP with evolving thresholds capable of dealing with databases with continuous data is proposed. This method is able to deal with continuous values directly, that is, without using any discretization method as a preprocessing step. Furthermore, the threshold values are evolved by mutation in every generation in order to obtain as many association rules as possible.

4.2 Literature review

The goal of association rule mining is to discover important associations among items. To achieve this purpose, Agrawal and his coworkers proposed several data mining algorithms based on the concept of large itemsets in order to find association rules in transaction data of databases consisting of only categorical attributes [4, 21, 22]. However, in most real life applications, databases contain many other attribute values besides 0 and 1. In order to deal with continuous values, the algorithms usually make use of discretization methods that transforms them into discrete attributes. Discretization is a process of transforming a continuous attribute value into a finite number of intervals and assigning each interval to a discrete numerical value [23, 24, 25, 26]. Discretization algorithms can be divided into two categories: unsupervised algorithms that discretize attributes without taking into account respective class labels. The representative algorithms are equal-width and equal-frequency discretization [27]. Supervised algorithms discretize attributes by taking into account the class-attribute interdependence. The representative algorithms are: maximum entropy [28], Patterson-Niblett algorithm [29], which is built-in as a front end into a decision trees algorithm [30], statistics based algorithms like ChiMerge [31] or Chi-squared [32] and class-attribute interdependency algorithms like CADD algorithm [33]. The disadvantage of the discretization algorithms described above is that the user must most often specify the number of intervals, or provide some heuristic rule to be used, then it is difficult to get the highest attribute interdependency and at the same time to get the lowest number of intervals.

However, alternative ways for handling continuous valued attributes are the induction of decision trees, by either generating a full tree by genetic programming operators or a heuristic method to generate the tree [34]. Another strategy is gener-
4.3 Association Rule Mining using GNP with Evolving Thresholds

In this chapter, the extraction of association rules from databases with continuous values using Genetic Network Programming with evolving thresholds is explained. Let $A_i$ be attribute (item) with continuous value in database and $a_i$ be its threshold. The proposed method extracts the following association rules:

$$(A_i > a_i) \land \cdots \land (A_j > a_j) \Rightarrow (A_m > a_m) \land \cdots \land (A_n > a_n).$$

Where one of the features of this method is to evolve the threshold $a_i$ along with the evolution of GNP.

4.3.1 Determining threshold values for the attributes

The estimation of threshold $a_i$ for a continuous attribute $A_i$ is explained as follows. At first, the mean $\mu_i$ and standard deviation $\sigma_i$ of every attribute $A_i$ is calculated, then an initial threshold value $a_i$ is selected randomly from the interval $[\mu_i - \alpha_i \sigma_i, \mu_i + \alpha_i \sigma_i]$, where $\alpha_i$ is a parameter that determines the size of the interval. The above is explained using an example. Figure 4.1(a) shows a database with continuous values. Thus, consider attribute $A_1$, the mean $\mu_1$ and standard deviation $\sigma_1$ are calculated, then according on the parameter $\alpha$, the upper and lower limits of the interval are calculated. In this example $\alpha = 1$. Once the interval is obtained, an initial threshold value is selected randomly from it, as is shown in Figure 4.1(b). Moreover, the initial threshold $a_i$ is evolved by mutation in every generation in order to obtain as many association rules as possible.

Once the threshold $a_i$ is selected for all attributes, each value of the attribute $A_i$ is checked if it is greater than the threshold $a_i$ in the judgment nodes of the GNP individuals. The evolution of the thresholds is carried out by introducing an additional parameter which determines the mutation rate $r$. Mutation rate $r$ is initialized...
Chapter 4. Association Rule Mining using Genetic Network Programming with Evolving Thresholds

(a) Calculate the mean and standard deviation of every attribute in the database.

\[ \mu_1 = 6 \]
\[ \sigma_1 = 2.64 \]

\[ \mu_2 = 141.67 \]
\[ \sigma_2 = 27.53 \]

\[ \mu_3 = 0.57 \]
\[ \sigma_3 = 0.41 \]

\[ \mu_4 = 31.67 \]
\[ \sigma_4 = 20.20 \]

(b) Calculate the limits of the interval.

Upper limit: \( \mu_i + (\alpha \times \sigma_i) \)

Lower limit: \( \mu_i - (\alpha \times \sigma_i) \)

\[ \alpha = 1 \]

\[ \mu_1 + (\alpha \times \sigma_1) = 6 + (1 \times 2.64) = 8.64 \]

Initial threshold is selected randomly from the interval

\[ a_1 = 4.70 \]

\[ \mu_1 - (\alpha \times \sigma_1) = 6 - (1 \times 2.64) = 3.35 \]

Figure 4.1: Estimation of the initial threshold values.

at 1/3 and as the generations go on, \( r \) is adjusted in order to select the best value. Two methods are proposed for adjusting mutation rate \( r \). The first one is to adjust mutation rate \( r \) by using the information on the number of association rules extracted in every generation. Then, if the number of association rules extracted increases in the current generation compared to the previous generation, mutation rate \( r \) is decreased by 0.01 because the current threshold \( a_i \) should be kept to generate more rules using the current threshold \( a_i \). On the other hand, if the number of association rules extracted decreases in the current generation compared to the previous generation, mutation rate \( r \) is increased by 0.01 because the current threshold \( a_i \) should be changed to generate more rules using different threshold \( a_i \). The second one is to adjust the mutation rate \( r \) gradually by selecting it randomly from the interval.
4.3. Association Rule Mining using GNP with Evolving Thresholds

\[ [r_c-0.02, r_c+0.02], \] where \( r_c \) is the current mutation rate. Table 4.1 includes the upper/lower limits of the intervals and initial threshold values in the case of \( \alpha_i = 1 \) for the database described in Figure 4.1(a).

Table 4.1: Upper and lower limits of threshold (\( \alpha_i=1.0 \))

<table>
<thead>
<tr>
<th></th>
<th>( A_1 )</th>
<th>( A_2 )</th>
<th>( A_3 )</th>
<th>( A_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>6.00</td>
<td>141.67</td>
<td>0.57</td>
<td>31.67</td>
</tr>
<tr>
<td>Std. deviation</td>
<td>2.64</td>
<td>27.53</td>
<td>0.41</td>
<td>20.20</td>
</tr>
<tr>
<td>Upper limit (mean + stdv)</td>
<td>8.64</td>
<td>169.20</td>
<td>0.98</td>
<td>51.87</td>
</tr>
<tr>
<td>Lower limit (mean - stdv)</td>
<td>3.35</td>
<td>114.14</td>
<td>0.16</td>
<td>11.47</td>
</tr>
<tr>
<td>Threshold interval</td>
<td>3.35 ( \leq a_1 \leq 8.64 )</td>
<td>114.14 ( \leq a_2 \leq 169.20 )</td>
<td>0.16 ( \leq a_3 \leq 0.98 )</td>
<td>11.47 ( \leq a_4 \leq 51.87 )</td>
</tr>
<tr>
<td>Threshold</td>
<td>4.70</td>
<td>117.30</td>
<td>0.10</td>
<td>17.90</td>
</tr>
</tbody>
</table>

4.3.2 GNP structure using threshold values

GNP examines the attribute values of database tuples using judgment nodes and calculates the measurements of association rules using processing nodes [6]. Attributes and their continuous values correspond to judgment nodes and their judgments in GNP, respectively. Therefore, the connections of judgment nodes are represented as association rules. Figures 4.2 and 4.3 show the structure of GNP with evolving threshold values for mining association rules. \( P_1 \) is a processing node and is a starting point of association rules. Yes-side of the judgment node is connected to another judgment node. Judgment nodes can be reused and shared with some other association rules because of GNP’s features. No-side of the judgment node is connected to the next numbered processing node. Judgment node determines the next node by a judgment result (Yes/No). In Figure 4.3, No-side of judgment nodes are abbreviated. For instance, from the database described in Figure 4.1(a) and Table 4.1, the threshold of the attribute \( A_1 \) is 4.70; and \( A_1 \) of tuple 1 in \( TID \) satisfies the condition of the threshold because \( A_1=8 \) in tuple 1; on the other hand, the threshold of the attribute \( A_2 \) is 117.30, however the attribute value of \( A_2 \) does not satisfy the condition of the threshold because \( A_2=115 \) in tuple 1, therefore the node transition from \( P_1 \) to \( P_2 \) occurs in Figure 4.2. If the examination of the connection from the starting point \( P_s \) ends, then
GNP examines tuple \( t \in TID \). Thus, all tuples in the database will be examined. The total number of tuples moving to Yes-side at each judgment node is calculated for every processing node, which is a starting point for calculating association rules. All GNP individuals are searched in parallel at the same time. If Yes-side connection of judgment nodes continues and the number of the judgment nodes becomes a cutoff value (maximum number of attributes in extracted association rules), then Yes-side connection is transferred to the next processing node obligatorily. In Figure 4.2, \( N \) is the number of total tuples, and \( a, b, c \) and \( d \) are the numbers of tuples moving to Yes-side at each judgment node. Table 4.2 shows the measurements such as support and confidence of association rules in Figure 4.2.

![GNP structure using threshold values.](image)

Figure 4.2: GNP structure using threshold values.

![Example of GNP individuals for extracting rules with threshold values.](image)

Figure 4.3: Example of GNP individuals for extracting rules with threshold values.
4.3.3 Calculation of the Support, Confidence and $\chi^2$ measurements using GNP

The proposed method measures the significance of association rules via support, confidence and $\chi^2$ value. Therefore, important association rules should satisfy the following:

$$\chi^2 > \chi^2_{\text{min}},$$  \hspace{1cm} (4.1)  
$$\text{support} \geq \text{sup}_{\text{min}},$$  \hspace{1cm} (4.2)  
$$\text{confidence} \geq \text{conf}_{\text{min}}.$$  \hspace{1cm} (4.3)

Where, $\chi^2_{\text{min}}$, $\text{sup}_{\text{min}}$ and $\text{conf}_{\text{min}}$ are the minimum $\chi^2$, support and confidence values given by the supervisors.

In statistics, $\chi^2$ test is a widely used method for testing the independence or correlation among variables. Basically, $\chi^2$ test is based on the comparison of observed frequencies with the corresponding expected frequencies. The closer the observed frequencies are to the expected frequencies, the larger the evidence in favor of independence is. If all the variables are in fact independent, the $\chi^2$ value would be 0. If it is higher than a cutoff value (6.63 at the 99% significance level), the independence assumption is rejected.

In the past, many other successful techniques utilized the $\chi^2$ test for independence or correlation [36, 37, 38]. In the proposed algorithm, $\chi^2$ test is used for generating correlated association rules. By adding a correlation measure to the support-confidence framework, the discovery of more meaningful correlated rules becomes possible.

The $\chi^2$ statistic is defined as follows:

$$\chi^2 = \sum_{\text{AllCells}} \frac{(O - E)^2}{E}. \hspace{1cm} (4.4)$$

Where, $E$ denotes the value of the expectation under the assumption of independence and $O$ the value of the observation. Then, let $\text{support}(X) = x$, $\text{support}(Y) = y$, $\text{support}(X \cup Y) = z$ and the number of database tuples equal $N$. If events $X$ and $Y$ are independent, then $\text{support}(X \cup Y) = xy$. $\chi^2$ is calculated using $x$, $y$, $z$ and $N$ as follows [7]:
\[ \chi^2 = \frac{N(z - xy)^2}{xy(1 - x)(1 - y)}. \] (4.5)

Table 4.2 shows how support and confidence of the association rules are calculated using \(a, b, c\) and \(d\) in Figure 4.2.

<table>
<thead>
<tr>
<th>association rules</th>
<th>support</th>
<th>confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A_1 \Rightarrow A_2)</td>
<td>(b/N)</td>
<td>(b/a)</td>
</tr>
<tr>
<td>(A_1 \Rightarrow A_2 \land A_3)</td>
<td>(c/N)</td>
<td>(c/a)</td>
</tr>
<tr>
<td>(A_1 \Rightarrow A_2 \land A_3 \land A_4)</td>
<td>(d/N)</td>
<td>(d/a)</td>
</tr>
<tr>
<td>(A_1 \land A_2 \Rightarrow A_3)</td>
<td>(c/N)</td>
<td>(c/b)</td>
</tr>
<tr>
<td>(A_1 \land A_2 \Rightarrow A_3 \land A_4)</td>
<td>(d/N)</td>
<td>(d/b)</td>
</tr>
<tr>
<td>(A_1 \land A_2 \land A_3 \Rightarrow A_4)</td>
<td>(d/N)</td>
<td>(d/c)</td>
</tr>
</tbody>
</table>

The extracted association rules are stored in a pool all together through generations. When an important rule is extracted by GNP, the overlap of the attributes is checked and it is also checked whether an important rule is new or not, i.e., whether it is already in the pool or not. If the rule is new, it is stored in the pool with its support, confidence and \(\chi^2\). If the association rule is not new, but the thresholds of items are different from the ones of the rule in the pool, the association rule with higher \(\chi^2\) value is stored, then the pool is updated in every generation and only important association rules with higher \(\chi^2\) values are stored.

### 4.3.4 Fitness of GNP

The evaluation of the individuals is made according to their fitness values. Then, reproduction of the individuals are carried out using genetic operations in order to generate a population for the next generation. This process is repeated until the last generation. The fitness of a GNP individual is computed by the following formula:

\[ F = \sum_{r \in R} \{ \chi^2(r) + 10(n_{ante}(r) - 1) + 10(n_{con}(r) - 1) + \alpha_{new}(r) \}, \] (4.6)

where,

\(R\): set of suffixes of extracted important association rules satisfying (4.1), (4.2) and (4.3)
4.4 Simulations

In a GNP individual.

$\chi^2(r)$: $\chi^2$ value of the association rule $r$.

$n_{ante}(r)$: the number of attributes in the antecedent of rule $r$.

$n_{con}(r)$: the number of attributes in the consequent of rule $r$.

$\alpha_{new}(r)$: additional constant defined by

$$
\alpha_{new}(r) = \begin{cases} 
\alpha_{new} & \text{ (rule } r \text{ is new)} \\
0 & \text{ (rule } r \text{ has been already extracted)}
\end{cases} 
$$

(4.7)

$\chi^2(r)$, $n_{ante}(r)$, $n_{con}(r)$ and $\alpha_{new}(r)$ are concerned with the importance, complexity and novelty of rule $r$, respectively.

Figure 4.4 shows the flowchart of the proposed association rule mining method.

Figure 4.4: Flowchart of the GNP-based association rule mining method.

4.4 Simulations

In order to confirm the effectiveness of the proposed method, the following simulations were executed:

- Simulation 1 discusses the results of fixed thresholds, they were set up at mean value $\mu_i$ at the first generation and remain fixed for all generations.
Chapter 4. Association Rule Mining using Genetic Network Programming with Evolving Thresholds

- Simulation 2 discusses the results where thresholds evolve in every generation, that is, a different threshold is randomly selected in each generation from the interval $[\mu_i - \alpha_i \sigma_i, \mu_i + \alpha_i \sigma_i]$.

- In simulations 3 and 4, mutation rate $r$ is introduced for determining whether the evolution of the thresholds is carried out.

- Simulation 3 discusses the results adjusting mutation rate $r$ by the information on the number of association rules extracted in every generation.

- Simulation 4 discusses the results adjusting mutation rate $r$ gradually by selecting it randomly from the interval $[r_c-0.02, r_c+0.02]$, where $r_c$ is the current mutation rate.

Simulations 2, 3 and 4 were done using different intervals of thresholds, that is $[\mu_i - \alpha_i \sigma_i, \mu_i + \alpha_i \sigma_i]$, $\alpha_i=1/6, 1/5, 1/4, 1/3, 1/2, 1, 2$ and $3$.

All experiments were run on a real database [39] that contains continuous attributes about heart disease. It consists in 50 attributes ($A_i$, $i=1,2,\ldots,50$) and 115 tuples. In simulations, the population size is 120. The number of processing nodes and judgment nodes are 10 and 78, respectively. ($\chi^2 = 6.63$), ($\sup_{\min} = 0.1$), and ($\alpha_{new} = 150$) were used. The detailed conditions of extracting association rules in the simulations are as follows: $n_{ante}(i) + n_{con}(i) \geq 6$, $n_{ante}(i) \leq 5$, $n_{con}(i) \leq 5$. The probability of crossover and mutation are $P_c = 15/78$, $P_{m1} = 1/3$ and $P_{m2} = 1/5$ (78 corresponds to the number of judgment nodes). The number of changing the connections of the processing nodes at each generation is 5. The simulations were executed for 500 generations and 20 trials are studied here for all the experiments changing the random sequences. All algorithms were coded in Java. Experiments were done on a 1.50GHz Pentium M with 504MB RAM.

Figure 4.5 and Figure 4.6 show the number of association rules extracted in the pool and the average fitness values of the GNP individuals using fixed thresholds in simulation 1, respectively.

Figure 4.7 shows the number of association rules extracted in the pool using different intervals in simulation 2. According to the experiments, it is found that the most suitable value for the interval is $\alpha_i = 1/6$ because higher and lower values than $\alpha_i = 1/6$ generates less amount of association rules. In other words, it is found from Figure 4.7 that the number of rules extracted decreases when the interval is larger
than $\alpha_i = 1/6$ or when the interval is smaller than $\alpha_i = 1/6$ (Although the case of smaller than $\alpha_i = 1/6$ is not described in Figure 4.7, $\alpha_i = 1/6$ has been confirmed as the best value). The above describes that a too small interval does not produce a large number of rules because the value of threshold does not change so much and is near to the mean and also it is confirmed that a too large interval does not produce a large number of rules as well because the value of threshold does change so violently. Figure 4.8 shows the average fitness value in the case of $\alpha_i = 1/6$. Compared with Figure 4.6, the fitness value became worse because the threshold values change considerably in every generation.

Figure 4.9 shows the number of rules extracted in the pool in simulation 3. It is found that the number of rules extracted has been increased for all $\alpha_i$ of the intervals because mutation rate $r$ is introduced and it makes the evolution of some thresholds occur less frequently as is explained in Simulation 2. It shows the fact that the constant change of the threshold values at all generations is harmful for extracting many
rules. Figure 4.10 also shows that the average fitness value is increased compared to simulation 2. The most suitable $\alpha_i$ value for the interval is also $\alpha_i=1/6$ because higher and lower values of $\alpha_i$ generates less amount of association rules as is explained in simulation 2. Figure 4.11 shows that mutation rate $r$ changes up and down at an early generations and approaches a certain value at the latter generations. This is because at an earlier generations when a fairly number of rules are generated, then mutation rate $r$ is decreased in order to generate more rules by keeping the threshold value, on the other hand, when the rules are no extracted, then mutation rate $r$ is increased in order to generate more rules by changing the threshold value.

Figure 4.12 shows the number of rules extracted in the pool in simulation 4. It is found from Figure 4.12 that the number of rules extracted has been increased for all $\alpha_i$ of the intervals compared to simulation 3, which means simulation 4 outperforms
4.4. Simulations

Figure 4.9: Number of rules extracted in the pool in Simulation 3.

Figure 4.10: Average fitness values of GNP in Simulation 3.

Figure 4.11: Evolution of mutation rate in Simulation 3.
Chapter 4. Association Rule Mining using Genetic Network Programming with Evolving Thresholds

Figure 4.12: Number of rules extracted in the pool in Simulation 4.

Figure 4.13: Average fitness values of GNP in Simulation 4.

simulation 1, simulation 2 and also simulation 3. Figure 4.13 shows that the average fitness value is also increased compared to other simulations, because the threshold change is not so drastic. Figure 4.14 shows that the evolution of mutation rate $r$ changes fairly, which shows the fact that a reasonable frequent change of the threshold values is helpful for extracting many rules. The most suitable $\alpha_i$ value for the interval is also $\alpha_i = 1/6$ because higher and lower values than $\alpha_i = 1/6$ generates less amount of association rules as is explained in simulations 2 and 3.
4.5 Summary

Association rules mining for continuous attributes has been a research topic for a long time. The GNP-based data mining method with evolving thresholds proposed in this chapter is one solution of this. Experiments were performed and the performances of the proposed algorithm were estimated. The results have shown that the GNP-based method extracts important association rules in the database effectively without using any discretization method as a preprocessing step and is able to evolve both GNP and threshold parameters concurrently.

Therefore, the main contributions of this work is described as follows:

- GNP was able to extract a large number of rules without the bloating problem usually found in other evolutionary algorithms such as Genetic Programming. Using the GNP’s structure, there is no need to explicitly encode the association rules into the genome of the GNP individual, instead, GNP individuals are just used to extract association rules through generations.

- A correlation measure was employed to enhance the support-confidence framework, where the rule is measured not only by its support and confidence but also by a correlation measure. Among many different correlation measures, the $\chi^2$ test for independence is one of the most important because of its solid statistical basis.

- Taking the GNP’s structure into account, the extraction of association rules was done without identifying frequent itemsets as most Apriori-based data mining...
algorithms. The frequent item set searching would be the bottleneck of the association rule mining algorithms due to its long searching time.

- Extracted association rules are stored in a pool all together through generations. The pool of association rules is updated in every generation exchanging the association rule with higher $\chi^2$ value for the same association rule with lower $\chi^2$ value.

- It was found the most suitable value for determining the interval in order to select the threshold values. That is, small intervals do not produce a large number of rules because the value of threshold does not change so much and is near to the mean value and it was also confirmed that large intervals do not produce a large number of rules as well because the value of threshold changes so drastically.

- It was found that the evolution of the threshold values as the generations go on is harmful for extracting many rules. Thus, the evolution of thresholds is executed at earlier generations and approaches a certain value at the latter generations. This is because at earlier generations when a fairly number of rules are generated, the mutation rate is decreased in order to generate more rules by keeping the threshold value, on the other hand, when there is no extraction of association rules the mutation is increased in order to generate more rules by changing the threshold value. Generally, at latter generations the number of rules extracted is small, then the mutation rate is increased to a certain value, then more rules are obtained.
Chapter 5

Mining Fuzzy Association Rules using Genetic Network Programming

5.1 Introduction

The initiative of combining association rule mining with fuzzy set theory has been applied more frequently in recent years. The original idea comes from dealing with continuous attributes in a database, where discretization of the continuous values into intervals would lead to under or overestimate the values that are near the borders. This is called the sharp boundary problem. Fuzzy sets can help us to overcome this problem by allowing different degrees of the membership, not only 1 and 0 treated by traditional methods. Attributes values can thereby be the members of more than one set and therefore give a more realistic view on such data. On the other hand, fuzzy set theory has been shown to be a very useful tool in association rule mining, because the mined rules can be expressed in linguistic terms, which are more natural and understandable for human beings. The linguistic representation is mainly useful when those discovered rules are presented to human experts for study.

In the literature, several techniques have been proposed on the use of evolutionary algorithms for discovering fuzzy association rules [40, 41, 42, 43]. The motivation for applying evolutionary algorithms to data mining lays in they are robust search methods which perform a global search in the space of candidate solutions [5].

In this chapter, a novel association rule mining approach that integrates Genetic
Network Programming and fuzzy set theory is proposed for mining interesting fuzzy rules from given quantitative data. The main characteristics of the proposed method are: 1) taking the GNP’s structure into account and the extraction of fuzzy association rules is done without identifying frequent itemsets, 2) it is based on the evolution of fuzzy association rule sets, where the parameters of the fuzzy membership functions are evolved and they become different rule by rule. As a result, more effective and stronger rules are extracted according to the progress of the evolution, 3) there is no need to explicitly encode the fuzzy association rule into the genome of the GNP individual, instead, GNP individuals are used just to extract fuzzy rules. 4) GNP’s structure allows to calculate the support, confidence and \( \chi^2 \) values easily in order to quantify the significance of the rules to be integrated into the model. 5) the fuzzy rules are mined through generations and stored in a general pool. Whereas, generally in the conventional algorithms: 1) the extraction of the rules is made identifying the frequent itemsets, particularly used in most Apriori-based data mining algorithms, 2) the parameters of the fuzzy membership functions are the same for all fuzzy rules extracted, 3) each individual represents the fuzzy rules encoding the rule itself into the individuals, 4) support and confidence framework is used as a measurement of the rules, and 5) the final result of the evolutionary process is the set of individuals.

5.2 Literature review

5.2.1 Mining Fuzzy Association Rules

In our understanding, the first paper introducing fuzzy sets into association rules was proposed by Lee and Kwang in 1997 [44]. The model uses a membership threshold to change fuzzy transactions into crisp ones before looking for ordinary association rules in the set of crisp transactions. Au and Chan proposed the F-APACS method that uses an adjusted difference analysis to identify interesting associations between attributes [45, 46]. Kuok et al. proposed an approach where the usefulness of itemsets and rules is measured by means of a significance factor, defined as a generalization of support based on sigma-counts [47]. In the method proposed by Hong et al. each quantitative value is transformed into a fuzzy sets in linguistic terms [48]. The algorithm then calculates the scalar cardinalities of all linguistic terms in the transaction data, therefore the method is focused on the most important linguistic terms, reducing its time complexity. A different attempt has been made by Tzung-Pei et al.,
which similarly uses the Apriori algorithm as a basis but incorporates fuzzy sets for mining quantitative values in a database [49]. Yue et al. proposed an approach to find fuzzy association rules with weighted items from transaction data [50]. Each item was given a weight to represent the importance of an item, and each weight was in a range of [0, 1]. They also adopted Kohonen self-organized mapping to derive fuzzy sets for numerical attributes. Weighted supports and confidences were utilized to discover weighted fuzzy association rules. Gyenesei presented another Apriori-like approach [51]. The algorithm first searches the database and returns the complete set containing all attributes of the database. In the second step, a transformed fuzzy database is created from the original one. After generating the candidate itemsets, the transformed database is scanned in order to evaluate the support and the items with a too low support are deleted. Zhang et al. presented an extension of the Equi-depth (EDP) algorithm introduced by Srikant and Agrawal [52] for mining fuzzy association rules involving quantitative attributes. The approach combines the obtained partitions with predefined linguistic labels [53]. Hu et al. proposed an effective algorithm named Fuzzy Grid Based Rules Mining Algorithm (FGBRMA) that generates large fuzzy grids and fuzzy association rules by using boolean operations on suitable table data structures [41]. Related with the above mentioned approach is the methodology by Fu et al. that finds the fuzzy sets to represent suitable linguistic labels for data by using fuzzy clustering techniques [54]. In this method, the user does not need to define the support and confidence values, that can be an advantage, however the obtained fuzzy sets would be hard to fit to meaningful labels. Miller and Yang proposed a distance-based association rules mining process, which improves the semantics of the intervals [55]. Hirota and Pedrycz proposed a context sensitive fuzzy clustering method based on fuzzy C-means to construct rule-based models [56]. Ishibuchi and E. Hullermeier illustrated fuzzy versions of confidence and support [57, 58].

### 5.2.2 Mining both Fuzzy Membership Functions and Fuzzy Association Rules

The above proposed approaches mine fuzzy rules under a given set of membership functions. The given membership functions had a critical influence on the final mining results. Recently, evolutionary algorithms such as Genetic algorithms (GAs) and Genetic Programming (GP) have also been used in the field of fuzzy data-mining algorithms for extracting both association rules and membership functions from quantita-
tive transactions. For instance, Wang et al. tuned membership functions for intrusion
detection systems based on similarity of association rules [59]. M. Kaya proposed a
GA-based clustering method to derive a predefined number of membership functions
for getting a maximum profit [60]. Hong et. al. proposed a GA-based fuzzy data-
mining method for extracting both association rules and membership functions from
quantitative transactions [61]. Mendez et. al. proposed a co-evolutionary system for
discovering fuzzy classification rules. The system uses two evolutionary algorithms:
a GP algorithm evolving a population of fuzzy rule sets and a simple evolutionary
algorithm evolving a population of fuzzy membership functions. The two populations
coe-volve, so that the final result is a fuzzy rule set and a set of membership functions
that are well adapted to each other [40].

The proposed GNP-based fuzzy mining approach lays into these kind of algo-
rithms since fuzzy rules and membership functions are dynamically mined, modifying
the parameters of the membership functions generation by generation. The main
differences between the above algorithms and the proposed algorithm are: (1) there
is no need to explicitly encode the fuzzy association rule into the genome of the GNP
individual, instead, GNP individuals are used just to extract the fuzzy rules. (2) Tak-
ing the GNP’s structure into account, a large number of rules are extracted without
the bloating problem in GNP individuals. (3) Our model is based on the evolution
of fuzzy association rule sets, where the parameters of the fuzzy membership func-
tions become different rule by rule. (4) The mined rules that satisfy the minimum
thresholds for significance measures are stored in a pool generation by generation,
whereas in the above algorithms the final result of the evolution process is the set of
individuals.

5.3 Mining Fuzzy Association Rules using GNP

This section describes the association rule mining algorithm based on Genetic Network
Programming and fuzzy set theory. The mined fuzzy association rules have the form

\[ \text{IF } [\text{conditions}] \text{ THEN [prediction]} \]

as follows:

\[ \text{If } (A_i \text{ is } Q_i) \land \cdots \land (A_j \text{ is } Q_j) \Rightarrow (A_m \text{ is } Q_m \land \cdots \land (A_n \text{ is } Q_n) \]

where, \(Q_i, Q_j, \ldots, Q_m\) and \(Q_n\) are the linguistic terms of the fuzzy attributes \(A_i, A_j, \ldots, A_m\)
and \(A_n\).

For instance: If the value of \(A_1\) is High \(\land A_2\) is Middle \(\Rightarrow A_7\) is Low is included in
the above form.

At first, a fuzzy discretization technique is used for dealing with databases with continuous attributes. Applying an association rule mining to the obtained fuzzy dataset will help to overcome the sharp boundary problem, besides to obtain a set of fuzzy association rules which are more understandable by domain experts. Then, the extraction of fuzzy association rules using GNP individuals is done. The measurements of the fuzzy rules are easily calculated by GNP and the fuzzy membership values are used for the probabilities for fuzzy rules extraction. Then, extracted fuzzy rules are stored in a general pool generation by generation. Moreover, the evolution of the fuzzy membership functions by using non-uniform mutation is done. Non-uniform mutation is used to perform a more global search in the space of candidate membership functions. Finally, updating the pool of rules is done according to the $\chi^2$ value of the new extracted fuzzy rules.

5.3.1 Handling continuous attributes using Fuzzy Sets

At first, the values of all continuous attributes of the database are fuzzified into three linguistic values namely low, middle, and high as Figure 5.1 shows. These linguistic values are defined by the combination of two trapezoidal and one triangular membership functions symmetrically spaced. Each continuous attribute is associated with its own membership functions. The parameters $\alpha$, $\beta$, and $\gamma$ collectively define the three membership functions (low, medium, and high) for the corresponding attribute, as shown in Figure 5.1. This representation enforces that, for each original value of the continuous attribute, the sum of its degrees of membership into the three linguistic values will be 1, which is intuitively sensible.

![Figure 5.1: Membership functions for attribute $i$.](image-url)
Figure 5.2 shows an example of the fuzzy discretization method. That is, Figure 5.2(a) shows how from a small database that is composed of Age and Salary attributes, a dataset with fuzzy attribute values is obtained by using the fuzzy membership functions and parameters described in Figure 5.2(b). The fuzzy values will be used to determine the transition between judgment nodes in GNP and extraction of fuzzy rules, which will be explained in the next section.

(a) Fuzzy attributes obtained from databases with continuous values.

(b) Definition of the membership functions.

Figure 5.2: The proposed fuzzy discretization technique.
Moreover, the parameters $\alpha$, $\beta$ and $\gamma$ defining the membership functions are evolved generation by generation in order to discover more interesting rules. Non-uniform mutation [62] was used for the evolution of the parameters of the membership functions. This genetic operator which reduces the disadvantages of random mutation, allows shifting the coordinates of the trapezoid and triangular, performing a more global search in the space of candidate membership functions. The parameters $\alpha$, $\beta$ and $\gamma$ are predefined only for the first generation and from the second generation, $\alpha$ and $\beta$ evolve by non-uniform mutation and $\gamma$ is automatically calculated by $(2\beta - \alpha)$ in order to maintain parameters symmetrically spaced. Figure 5.3 shows an example of the parameters and fuzzy membership functions evolution according to the generation goes on.

![Figure 5.3: Example of the evolution of the fuzzy parameters.](image)

### 5.3.2 GNP structure for mining fuzzy association rules

The basic structure of GNP individuals for fuzzy association rule mining is shown in Figure 5.4. The fuzzy rules are represented as the connections of judgment nodes in GNP. Each judgment node examines the fuzzy attributes, which were calculated in the last section [See Figure 5.2(a)]. $P_1$ is a processing node and is a starting point of
extracting association rules. Each processing node has an inherent numeric order \((P_1, P_2, \ldots, P_s)\) and is connected to a judgment node. Yes-side of the judgment node is connected to another judgment node. No-side of the judgment node is connected to the next numbered processing node. In order to calculate the support and confidence of the fuzzy rules, the total number of tuples moving to Yes-side at each judgment node is calculated for every processing node. In Figure 5.4, \(N\) is the total number of tuples and \(a, b, c\) and \(d\) are the number of tuples moving to Yes-side at each judgment node. These counts are used for the calculation of the measurements.

\[
\begin{align*}
(A_1 \text{ High}) \land (A_2 \text{ Low}) \land (A_3 \text{ Mid}) & \Rightarrow (A_4 \text{ Low}) \\
(A_1 \text{ High}) \land (A_2 \text{ Low}) & \Rightarrow (A_3 \text{ Mid}) \land (A_4 \text{ Low}) \\
(A_1 \text{ High}) & \Rightarrow (A_2 \text{ Low}) \land (A_3 \text{ Mid}) \land (A_4 \text{ Low})
\end{align*}
\]

Figure 5.4: Basic GNP structure for fuzzy association rule mining.

Once a GNP individual starts the searching for association rules, the fuzzy values are employed to determine the transition from one judgment node to another, that is, the fuzzy value is used as the probability for moving to the Yes-side or No-side of the judgment node. Figure 5.5 shows how the transition of judgment nodes is done using probability \(P_t\) determined by the fuzzy values. In Figure 5.5, \(r\) is a random variable in \([0,1]\), \(a_i\) is the value of fuzzy attribute \(A_i\), \(\mu_{Q_i}(a_i)\) is the value of the membership function \(\mu_{Q_i}(A_i)\) when the value of fuzzy attribute \(A_i\) is \(a_i\). A random number is generated and compared to the membership value of the fuzzy attribute. If the random number is smaller than or equal to the membership value, then go to the Yes-side of the judgment node, otherwise, go to the No-side of the judgment node. Figure 5.6 shows an example of the transition of nodes using the fuzzy values and transition probability \(P_t\). Table 5.1 shows how to calculate the support and confidence of the fuzzy rules extracted from the GNP individual in Figure 5.4.

Calculation of \(\chi^2\) value, fitness function and execution of the genetic operators are the same as the method described in former chapter.
5.3. Mining Fuzzy Association Rules using GNP

Figure 5.5: Probabilistic transition from one judgment node to another.

Figure 5.6: Transition from one judgment node to another using fuzzy values using probability $P_t$. 

<table>
<thead>
<tr>
<th>TID</th>
<th>$A_{i_{High}}$</th>
<th>...</th>
<th>$A_{i_{Low}}$</th>
<th>...</th>
<th>$A_{i_{Mid}}$</th>
<th>...</th>
<th>$A_{i_{Low}}$</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8</td>
<td></td>
<td>0.7</td>
<td></td>
<td>0.75</td>
<td></td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.7</td>
<td></td>
<td>0.65</td>
<td></td>
<td>0.2</td>
<td></td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.85</td>
<td></td>
<td>0.1</td>
<td></td>
<td>0.8</td>
<td></td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.7</td>
<td></td>
<td>0.8</td>
<td></td>
<td>0.7</td>
<td></td>
<td>0.1</td>
<td></td>
</tr>
</tbody>
</table>

Fuzzy values are used as the probability $P_t$ moving to the Yes-side.
Table 5.1: Support and confidence of the fuzzy association rules.

<table>
<thead>
<tr>
<th>association rules</th>
<th>support</th>
<th>confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_{1,.High} \Rightarrow A_{2,.Low}$</td>
<td>b/N</td>
<td>b/a</td>
</tr>
<tr>
<td>$A_{1,.High} \Rightarrow A_{2,.Low} \land A_{3,.Mid}$</td>
<td>c/N</td>
<td>c/a</td>
</tr>
<tr>
<td>$A_{1,.High} \Rightarrow A_{2,.Low} \land A_{3,.Mid} \land A_{4,.Low}$</td>
<td>d/N</td>
<td>d/a</td>
</tr>
<tr>
<td>$A_{1,.High} \land A_{2,.Low} \Rightarrow A_{3,.Mid}$</td>
<td>c/N</td>
<td>c/b</td>
</tr>
<tr>
<td>$A_{1,.High} \land A_{2,.Low} \Rightarrow A_{3,.Mid} \land A_{4,.Low}$</td>
<td>d/N</td>
<td>d/b</td>
</tr>
<tr>
<td>$A_{1,.High} \land A_{2,.Low} \land A_{3,.Mid} \Rightarrow A_{4,.Low}$</td>
<td>d/N</td>
<td>d/c</td>
</tr>
</tbody>
</table>

5.3.3 Updating the pool of Fuzzy Rules

The mined fuzzy rules are stored in a pool through generations as Figure 5.7 shows. When an important rule is extracted by GNP, it is stored in a pool of rules with its support, confidence, $\chi^2$ values and fuzzy parameters. Occasionally, a fuzzy rule already stored in the pool can be once again extracted, in that situation, the membership functions and $\chi^2$ values might be changed. Then, only the fuzzy rule with higher $\chi^2$ value could replace the same old fuzzy rule in the pool along with its fuzzy parameters. Therefore, the pool is updated in every generation and only important fuzzy rules with higher $\chi^2$ values and better adapted fuzzy parameters are stored.

![Figure 5.7: Updating the pool of fuzzy rules in every generation.](image-url)
Figure 5.8 shows the framework for mining fuzzy membership functions and fuzzy association rules.

![Diagram of the framework](image)

Figure 5.8: Framework for searching fuzzy membership functions and fuzzy association rules.

### 5.3.4 Flow of the Fuzzy Association Rule Mining

The algorithm for discovering fuzzy rules from continuous data with probabilistic transitions is described as follows:

**INPUT:** A dataset with \( n \) continuous attributes, a set of fuzzy membership functions \( \mu_{Qn}(A_n) \), a predefined number of generations \( T \), a predefined minimum support \( (\text{sup}_{\text{min}}) \), minimum confidence \( (\text{conf}_{\text{min}}) \) and minimum \( \chi^2 \) \( (\chi^2_{\text{min}}) \) thresholds.

**OUTPUT:** A pool of fuzzy association rules with support, confidence and \( \chi^2 \) values larger than or equal to the predefined minimum support, confidence and \( \chi^2 \) thresholds.

**STEP 1:** Randomly generate a population of GNP individuals with a predefined number of judgment and processing nodes.

**STEP 2:** Fuzzify each continuous attribute of the dataset into three linguistic values, *e.g.* low, middle and high. Each continuous attribute is associated with its own membership functions.
STEP 3: Extract fuzzy association rules using GNP as follows:

STEP 3.1: Evaluate if a fuzzy attribute is high, medium or low using judgment nodes by the following: the transition from one judgment node to another is executed using the membership value. Concretely, a random number in [0.1] is generated and compared to the fuzzy membership value. If the random number is smaller than or equal to the fuzzy membership value, then go to the Yes-side of the judgment node, otherwise, go to the No-side of the judgment node.

STEP 3.2: Calculate the support, confidence and $\chi^2$ of the fuzzy association rule by counting the total number of tuples moving to Yes-side at each judgment node using the processing nodes.

STEP 4: Store the fuzzy association rules that satisfy the minimum support, confidence and $\chi^2$ thresholds.

STEP 5: Check whether an important rule is new or not (whether it is already in the pool or not) as follows:

STEP 5.1: If the fuzzy rule is new, store it in the pool with its support, confidence, $\chi^2$ and the parameters of the fuzzy membership functions.

STEP 5.2: If the fuzzy rule is not new, that is, the current fuzzy rule was already extracted in former generations, compare the $\chi^2$ values, then, the fuzzy rule with higher $\chi^2$ value is stored in the pool along with its membership functions. Therefore, the pool is updated in every generation and only the important fuzzy rules with higher $\chi^2$ values are stored along with their adapted fuzzy parameters.

STEP 6: If the number of generations $T$ reaches, then stop the algorithm, otherwise go to the next step.

STEP 7: Perform the evolution of the GNP individuals as follows:

STEP 7.1: Calculate the fitness of each GNP individual.

STEP 7.2: Select the top 1/3 GNP individuals according to their fitness values.

STEP 7.3: Execute the genetic operators to the selected GNP individuals in order to create the next population.

STEP 8: Evolve the parameters of the fuzzy membership functions using non-uniform mutation.

STEP 9: Go to STEP 2.
5.4 Simulations

To confirm the effectiveness of the proposed fuzzy association rule mining, eight public-domain datasets from UCI (University of California at Irvine) dataset repository have been selected [63]. Table 5.2 describes the datasets along with some related statistical information. 80% of the records of the database is used for the training set and 20% of the records is used for the test set. The number of GNP individulas in the population is 120. The number of processing and judgment nodes are 20 and 200, respectively. $\chi^2 = 6.63$, $sup_{min} = 0.01$ and $\alpha_{new} = 150$ are used. The probability of crossover is $P_c = 1/5$, mutation of the connection is $P_{m1} = 1/3$ and mutation of the function is $P_{m2} = 1/5$. All algorithms has been developed in a Java-based software development environment. Experiments were performed on a 1.50GHz Pentium M with 504MB RAM.

Table 5.2: UCI ML datasets used in the experiments.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Number of records</th>
<th>Number of attributes</th>
<th>Number of classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waveform</td>
<td>5000</td>
<td>22</td>
<td>3</td>
</tr>
<tr>
<td>CRX</td>
<td>690</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>351</td>
<td>34</td>
<td>2</td>
</tr>
<tr>
<td>Heart (statlog)</td>
<td>303</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>Pima</td>
<td>768</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>Pageblocks</td>
<td>5473</td>
<td>11</td>
<td>5</td>
</tr>
<tr>
<td>Glass</td>
<td>214</td>
<td>11</td>
<td>7</td>
</tr>
<tr>
<td>Iris</td>
<td>150</td>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>

Figure 5.9 shows the total number of fuzzy rules extracted when the maximun number of generations is set to 100. As an illustration, CRX database was selected for showing the number of extracted rules for different minimum support values and the processing time in Figure 5.10 and Figure 5.11, respectively.

In order to show an example of the evolution of the fuzzy membership functions, Figure 5.12 shows the evolution of $\alpha$, $\beta$ and $\gamma$ for one of the attributes of CRX database. These parameters mutate within a range and in final stage converge due to non-uniform mutation properties.

Table 5.3 shows how the fuzzy rules are updated according to the $\chi^2$ value of the rules. That is, fuzzy rules can be extracted in different generations and the fuzzy parameters $\alpha$, $\beta$ and $\gamma$ for each fuzzy attribute may be different, then the rule with
the highest $\chi^2$ value is stored with its fuzzy parameters updating the pool.

5.5 Summary

Fuzzy association mining has been regarded as a promising area for both researchers and practitioners due to its advantages in expressing natural language and coping with uncertainty of knowledge. Due to the fact that most databases contain con-
5.5. Summary

Continuous data, fuzzy association rule mining methods might achieve wide acceptance in the future. Compared with traditional association rules, fuzzy rules provide good linguistic explanation and can deal with both discrete and quantitative attributes. More concretely, fuzzy association rules still might be one of the focal points of research interest, where partitioning or clustering for quantitative data, would attract considerable attention as well as linguistic modification. In this chapter, a model based on Genetic Network Programming for fuzzy transactions has been introduced. Therefore, the main contribution of this work is described as follows:

Figure 5.10: Number of extracted fuzzy rules in CRX DB for different minimum support values.

Figure 5.11: Processing time for CRX DB for different minimum support values.
The proposed method can extract a large amount of interesting fuzzy rules in the selected databases, this is because the GNP individuals are used to extract the fuzzy rules and judgment nodes can be reused and shared with some other association rules because of GNP’s structure.

- The fuzzy discretization technique developed to deal with continuous datasets,
5.5. Summary

<table>
<thead>
<tr>
<th>Fuzzy rule</th>
<th>Generation</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\gamma$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\gamma$</th>
<th>$\chi^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_4 High \land A_6 Mid$</td>
<td>$5$</td>
<td>$2.1$</td>
<td>$3.2$</td>
<td>$4.3$</td>
<td>$1.9$</td>
<td>$7.5$</td>
<td>$13.1$</td>
<td>$11.6$</td>
</tr>
<tr>
<td>$A_4 High \land A_6 Mid$</td>
<td>$8$</td>
<td>$1.9$</td>
<td>$3.3$</td>
<td>$4.7$</td>
<td>$1.5$</td>
<td>$4.0$</td>
<td>$6.5$</td>
<td>$12.4$</td>
</tr>
<tr>
<td>$A_4 High \land A_6 Mid$</td>
<td>$12$</td>
<td>$2.7$</td>
<td>$3.9$</td>
<td>$5.1$</td>
<td>$1.2$</td>
<td>$5.8$</td>
<td>$10.4$</td>
<td>$15.3$</td>
</tr>
<tr>
<td>$A_2 Low \land A_5 Mid$</td>
<td>$9$</td>
<td>$1.9$</td>
<td>$3.6$</td>
<td>$5.3$</td>
<td>$0.3$</td>
<td>$10.9$</td>
<td>$21.5$</td>
<td>$7.2$</td>
</tr>
<tr>
<td>$A_2 Low \land A_5 Mid$</td>
<td>$27$</td>
<td>$0.1$</td>
<td>$8.1$</td>
<td>$16.67$</td>
<td>$1.9$</td>
<td>$6.5$</td>
<td>$11.0$</td>
<td>$9.3$</td>
</tr>
</tbody>
</table>

shows the significance in the rule’s extraction process since the result is a set of fuzzy association rules.

- It is easy to compute membership values, however the critical task is to find an appropriate membership functions. This is achieved by evolving the fuzzy parameters. In other words, the evolution of the fuzzy membership functions shows the effectiveness for discovering new fuzzy rules generation by generation.

- The fuzzy membership values are used for fuzzy rules extraction, where the parameters of the membership functions are evolved by non-uniform mutation in order to perform a more global search in the space of candidate membership functions.

- Support, confidence and $\chi^2$ values were calculated to quantify the significance of the rules, therefore, more effective rules are involved in the fuzzy mining model.

In order to evaluate the effectiveness of the extracted fuzzy rules, how the proposed association rule mining model can be applied to classification problems has been also studied in this thesis. Therefore, the simulation results for evaluating the usefulness of the fuzzy association rules will be shown in next chapter.
Chapter 6

Genetic Network Programming for Extracting and Applying Fuzzy Classification Rules

6.1 Introduction

Recently, extensive research has been carried out to develop enhanced classification methods. One of the understandable models used is the association rule-based classification, where the aim is to combine the advantages of both traditional classification and association rule discovery. A class association rule is generally expressed as IF-THEN rule, i.e., IF [term1 AND term2 AND ... ] THEN [class]. Each term of the antecedent is a pair of [attribute, value]. The consequent is the result of classification, that is, the class value of the attributes. The most important advantages of associative classifiers can be reviewed as follows: (1) Despite of the size of the training set, the training mechanism is fast. (2) The speed of the classification might be increased with a compact set of rules. (3) The interpretability of the classification model is significantly improved. (4) Former performance studies show that associative classifiers result in better accuracy in general.

In this chapter, the fuzzy association rule mining method proposed in chapter 5 is applied to the classification task of data mining. The proposed model consists of two major phases: (1) generating fuzzy class association rules by using GNP, (2) building a classifier model based on the extracted fuzzy rules. In the first phase, the task is to extract fuzzy class association rules using a GNP-based algorithm. Each run of the
algorithm discovers fuzzy rules for a single class, therefore the algorithm must run $K + 1$ times, where $K + 1$ is the number of classes. In the second phase, all of the generated fuzzy rules in the pool are used to predict the class of the test set. For each test data, the classifier computes the average matching degree between data $d$ and the rules in class $k$. Finally, the class with the highest average matching degree is assigned to data $d$. The accuracy rate on the test set is computed as the number of correctly classified test examples divided by the total number of test examples.

6.2 Class Association Rule Mining

In this phase, the main task is to extract the fuzzy class association rules from a fuzzified training set. The rule has the following form:

$\text{If } (A_i \text{ is } Q_i) \land \cdots \land (A_j \text{ is } Q_j) \Rightarrow (C = k),$

where, $Q_i, \ldots, Q_j$ are the linguistic terms of the fuzzy attributes $A_i, \ldots, A_j$ and $k$ is the class ($k = 0, 1, 2, \ldots, K$).

For instance, the following rule means that the current fuzzy attributes belong to class $k$ when the values of the three attributes $A_1, A_2$ and $A_3$ are high, low and middle, respectively:

$\text{If } (A_1 \text{ is High}) \land (A_2 \text{ is Low}) \land (A_3 \text{ is Middle}) \Rightarrow (C = k).$

6.2.1 Fuzzy Membership Functions for handling continuous Data

The values of all continuous attributes are fuzzified into three linguistic values and evolved in the same way as the fuzzy discretization technique described in the former chapter. However, in order to confirm the effectiveness of non-uniform mutation proposed before, fixed and uniform mutation are also studied in this chapter for the evolution of the parameters $\alpha, \beta$ and $\gamma$ that define the fuzzy membership functions (FMF).

**Fixed-predefined FMF**

The parameters $\alpha$ and $\beta$ for attribute $A_i$ are predefined by analyzing the distribution of the data, $\gamma$ is automatically calculated by $(2\beta - \alpha)$ and they kept the same through all generations.
Uniform mutation of the FMF

The parameters $\alpha$, $\beta$ and $\gamma$ are set initially as in the fixed-predefined FMF. In the next generation, the parameters are selected for mutation with probability $P_m$. Let $x_n$ be a parameter selected for mutation in the $n$th generation, then, its mutated variable $x'_n$ is calculated as follows:

\[
x'_n = \begin{cases} 
  x_n + (UB - x_o)/\text{generation}, & \text{if a random } \xi \text{ is 0;} \\
  x_n - (x_o - LB)/\text{generation}, & \text{if a random } \xi \text{ is 1.} 
\end{cases}
\] (6.1)

where, $LB$ and $UB$ are the lower and upper bounds of the variable $x_n$, $x_o$ is the initial value of the parameter in the first generation, \textit{generation} is the generation number and $\xi$ is a random binary value.

Non-Uniform mutation of the FMF

The parameters $\alpha$, $\beta$ and $\gamma$ are set initially as in the fixed-predefined FMF. In the next generation, the parameters are selected for non-uniform mutation with probability $P_m$. Let $x_n$ be a parameter selected for non-uniform mutation in the $n$th generation, then, its mutated variable $x'_n$ is calculated as follows:

\[
x'_n = \begin{cases} 
  x_n + \Delta(n, UB - x_n), & \text{if a random } \xi \text{ is 0;} \\
  x_n - \Delta(n, x_n - LB), & \text{if a random } \xi \text{ is 1.} 
\end{cases}
\] (6.2)

where, $LB$ and $UB$ are the lower and upper bounds of the variable $x_n$ and $\xi$ is a random binary value. The function $\Delta(n, y)$ returns a value in the range $[0, y]$ in such a way that $\Delta(n, y)$ approaches to zero as generation number $n$ increases. This property causes this operator to search the space uniformly initially (when $n$ is small), and very locally at later generations. This strategy increases the probability of generating a new number closer to its predecessor than a random choice at later generations.

The following function was used:

\[
\Delta(n, y) = y \cdot (1 - r^{(1 - T)n}),
\] (6.3)

where, $r$ is a uniform random number from $[0,1]$, $T$ is the maximal generation number, and $\eta$ is a system parameter determining the degree of dependency on the generation number. In this paper, $\eta = 5$. 
6.2.2 GNP for Fuzzy Class Association Rule Mining

The basic structure of GNP for the extraction of fuzzy class association rules is shown in Figure 6.1. The fuzzy rules are represented as the connections of judgment nodes in GNP. Each judgment node examines the fuzzy attribute values. $P_1$ is a processing node and is a starting point of extracting association rules. Yes-side of the judgment node is connected to another judgment node. No-side of the judgment node is connected to the next numbered processing node. In Figure 6.1, $N$ is the total number of tuples, $a$, $b$, $c$ and $d$ are the number of tuples moving to Yes-side at each judgment node and $a(k)$, $b(k)$, $c(k)$ and $d(k)$ are the number of tuples moving to Yes-side at each judgment node that satisfy class $k$. These counts are used for the calculation of the measurements such as support, confidence and $\chi^2$.

Once a GNP individual starts the searching for association rules, the transition from one judgment node to another is probabilistically determined by using the fuzzy values (explained in section 5.3.1). However, in order to confirm the effectiveness of probabilistic transitions in GNP, deterministic transitions are also studied in this chapter.

**GNP with probabilistic transitions**

Figure 6.2 and Figure 6.3 show how the transition of nodes is done using probability $P_t$ determined by the fuzzy values.
6.2. Class Association Rule Mining

Figure 6.2: Probabilistic transition from one judgment node to another.

In Figure 6.2, \( r \) is a random variable in \([0,1]\), \( a_i \) is the value of fuzzy attribute \( A_i \), \( \mu_{Q_i}(a_i) \) is the value of the membership function \( \mu_{Q_i}(A_i) \) when the value of fuzzy attribute \( A_i \) is \( a_i \). A random number is generated and compared to the membership value of the fuzzy attribute. If the random number is smaller than or equal to the membership value, then go to the Yes-side of the judgment node, otherwise, go to the No-side of the judgment node.

![Diagram](image1)

Figure 6.3: Probabilistic transition from one judgment node to another using probability \( P_i \) determined by fuzzy values.
Chapter 6. Genetic Network Programming for Extracting and Applying Fuzzy Classification Rules

GNP with deterministic transitions

Figure 6.4 and Figure 6.5 show how the deterministic transition of judgment nodes is done in GNP individuals. If the membership value of the fuzzy attribute is greater than or equal 0.5, then go to the Yes-side of the judgment node, otherwise, go to the No-side of the judgment node.

Figure 6.4: Deterministic transition from one judgment node to another.

<table>
<thead>
<tr>
<th>TID</th>
<th>$A_1$ Low</th>
<th>Mid</th>
<th>High</th>
<th>$A_2$ Low</th>
<th>Mid</th>
<th>High</th>
<th>$A_3$ Low</th>
<th>Mid</th>
<th>High</th>
<th>$A_4$ Low</th>
<th>Mid</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0.2</td>
<td>0.8</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0.6</td>
<td>0.4</td>
<td>0</td>
<td>0.6</td>
<td>0.4</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0.3</td>
<td>0.7</td>
<td>0</td>
<td>0.4</td>
<td>0.6</td>
<td>0</td>
<td>0.2</td>
<td>0.8</td>
<td>0</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0.3</td>
<td>0.7</td>
<td>0</td>
<td>0</td>
<td>0.3</td>
<td>0.7</td>
<td>0.4</td>
<td>0.6</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0.4</td>
<td>0.6</td>
<td>0.1</td>
<td>0.9</td>
<td>0</td>
<td>0.7</td>
<td>0.3</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
</tr>
</tbody>
</table>

The transition to the Yes-side is done if the fuzzy value is $\geq 0.5$.

Figure 6.5: Deterministic transition from one judgment node to another comparing the fuzzy value with 0.5.
6.2.3 Fitness function

The fitness of GNP for extracting fuzzy class association rules is defined as follows:

\[ F = \sum_{r \in R} \left\{ \chi^2(r) + 10(n(r) - 1) + \alpha_{\text{new}}(r) \right\}, \quad (6.4) \]

where,
- \( R \): set of suffixes of extracted important association rules satisfying the minimum thresholds for measurements in a GNP individual.
- \( \chi^2(r) \): \( \chi^2 \) value of rule \( r \).
- \( n(r) \): the number of attributes in the antecedent part of rule \( r \).
- \( \alpha_{\text{new}}(r) \): additional constant defined by

\[ \alpha_{\text{new}}(r) = \begin{cases} \alpha_{\text{new}} & \text{(rule } r \text{ is new)} \\ 0 & \text{(rule } r \text{ has been already extracted)} \end{cases} \quad (6.5) \]

Notice that in Eq.(6.4) the consequent term is not considered because GNP individuals are used to extract fuzzy class association rules, where the consequent is fixed. That is, the rule has the form: Attribute \( \Rightarrow \) class. Then, in each run of the algorithm all GNP individuals of the population are associated with the same class. Therefore, there is no need to explicitly take the consequent into account into the fitness function of a GNP individual.

6.2.4 Calculation of measurements using GNP

In order to calculate the support, confidence and \( \chi^2 \) of the fuzzy class association rules, the total number of tuples moving to Yes-side at each judgment node is calculated for every processing node. In Table 6.1, \( N \) is the total number of tuples, \( a, b, c \) and \( d \) are the number of tuples moving to Yes-side at each judgment node and \( a(k), b(k), c(k) \) and \( d(k) \) are the number of tuples moving to Yes-side at each judgment node that satisfy class \( k \). Table 6.1 shows how to calculate the support and confidence of the fuzzy rules extracted from the GNP individual in Figure 6.1.

The mined fuzzy class association rules that satisfy the minimum support, confidence and \( \chi^2 \) thresholds are stored in a pool with its significance measures and fuzzy parameters through generations. Updating the pool is the same as the model described in chapter 5.
Table 6.1: Support and confidence of the fuzzy class association rules.

<table>
<thead>
<tr>
<th>association rules</th>
<th>supp.</th>
<th>conf.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_{1,High} \Rightarrow C = k )</td>
<td>( a(k)/N )</td>
<td>( a(k)/a )</td>
</tr>
<tr>
<td>( A_{1,High} \land A_{2,Low} \Rightarrow C = k )</td>
<td>( b(k)/N )</td>
<td>( b(k)/b )</td>
</tr>
<tr>
<td>( A_{1,High} \land A_{2,Low} \land A_{3,Mid} \Rightarrow C = k )</td>
<td>( c(k)/N )</td>
<td>( c(k)/c )</td>
</tr>
<tr>
<td>( A_{1,High} \land A_{2,Low} \land A_{3,Mid} \land A_{4,Low} \Rightarrow C = k )</td>
<td>( d(k)/N )</td>
<td>( d(k)/d )</td>
</tr>
</tbody>
</table>

### 6.2.5 Selection and Genetic Operators in GNP

Elite selection method is used. Once all crossovers and mutations have been performed, all the produced GNP offspring are sorted by their fitness values and only 1/3 best individuals are selected for reproduction. Crossover, Mutation-1 and Mutation-2 are executed to GNP individuals. Due to the genetic operators, new attributes are evaluated and new relationships between attributes are found in each generation, as a result, the new rules that satisfy the minimum thresholds for the significance measures are mined [See Figure 6.6 and Figure 6.7].

![Figure 6.6](image1.png)

**Figure 6.6:** The connection of the judgment nodes is changed by mutation rate of \( P_{m1} \).

![Figure 6.7](image2.png)

**Figure 6.7:** The function of the judgment nodes is changed by mutation rate \( P_{m2} \).
6.3 Building a classifier model

As mentioned above, a complete execution of the proposed fuzzy class association rule mining generates a pool of fuzzy association rules for each class in the dataset. Since every fuzzy rule is stored in the pool with its own fuzzy parameters, the attributes of data $d$ have to be fuzzified according to the fuzzy parameters of rule $r$. Then, these rules in the pool are used to predict the class of data $d$. For each test data, the classifier computes the average matching degree between data $d$ and the rules in class $k$. Finally, the class with the highest average matching degree is assigned to data $d$. The accuracy rate on the test set is computed as the number of correctly classified test samples divided by the total number of test samples.

Therefore, the classification of test data $d$ is determined as follows:

1) $R_k$: Pick up the set of suffixes of rules in class $k$, ($k = 0, 1, 2 \ldots K$).
2) $m_k(d)$: Compute the average matching degree between data $d$ and the rules in class $k$.

$$m_k(d) = \frac{1}{|R_k|} \sum_{r \in R_k} \text{Match}_k(d, r),$$  \hspace{1cm} (6.6)

$$\text{Match}_k(d, r) = \frac{N_k(d, r)}{N_k(r)},$$  \hspace{1cm} (6.7)

where,

$\text{Match}_k(d, r)$: matching degree between data $d$ and rule $r$ in class $k$.

$N_k(d, r)$: the sum of the fuzzy membership values of the fuzzy attributes in the antecedent part of fuzzy rule $r$ in class $k$, which are calculated by data $d$.

$N_k(r)$: the number of attributes in the antecedent part of rule $r$ in class $k$.

3) Predict in such way that data $d$ belongs to the class having the highest $m_k(d)$.

### 6.4 Simulations

To evaluate the accuracy and performances of the proposed fuzzy association rule mining and GNP-based classifier, eight public-domain datasets from UCI dataset repository have been selected [63]. Table 6.2 describes the datasets along with some related statistical information. There are two reasons why these eight classification problems were selected. (1) The datasets are multiclass pattern classification problems involving continuous attributes. (2) In order to compare the proposed method to existing algorithms. All algorithms have been developed in a Java-based software development environment. Experiments were performed on a 1.50GHz Pentium M with 504MB RAM.

To obtain the accurate performances of the proposed algorithm, the following were studied:

- GNP-based fuzzy rule mining with fixed and evolved fuzzy parameters.

- GNP-based fuzzy rule mining with deterministic and probabilistic node transitions.

Parameters setting of GNP are the same for all the simulations and are summarized in Table 6.3.
6.4. Simulations

Table 6.2: Multi-class datasets used in the experiments.

<table>
<thead>
<tr>
<th>Datasets</th>
<th># Record</th>
<th># Attribute</th>
<th># Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heart</td>
<td>303</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>351</td>
<td>34</td>
<td>2</td>
</tr>
<tr>
<td>CRX</td>
<td>690</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>Iris</td>
<td>150</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Glass</td>
<td>214</td>
<td>11</td>
<td>7</td>
</tr>
<tr>
<td>Pageblocks</td>
<td>5473</td>
<td>11</td>
<td>5</td>
</tr>
<tr>
<td>Waveform</td>
<td>5000</td>
<td>22</td>
<td>3</td>
</tr>
<tr>
<td>Pima</td>
<td>768</td>
<td>9</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 6.3: Parameters for GNP-based fuzzy class association rule mining.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>120</td>
</tr>
<tr>
<td>Maximum generations</td>
<td>100</td>
</tr>
<tr>
<td>Number of processing nodes</td>
<td>20</td>
</tr>
<tr>
<td>Number of judgement nodes</td>
<td>200</td>
</tr>
<tr>
<td>Crossover rate</td>
<td>1/5</td>
</tr>
<tr>
<td>Mutation rate of the connection</td>
<td>1/3</td>
</tr>
<tr>
<td>Mutation rate of the function</td>
<td>1/5</td>
</tr>
<tr>
<td>Minimum support</td>
<td>0.01</td>
</tr>
<tr>
<td>Minimum confidence</td>
<td>0.3</td>
</tr>
<tr>
<td>Minimum $\chi^2$</td>
<td>6.63</td>
</tr>
<tr>
<td>$\alpha_{new}$</td>
<td>150</td>
</tr>
</tbody>
</table>

The number of extracted fuzzy class association rules and accuracy obtained are presented from Figure 6.9 to Figure 6.16 and from Figure 6.17 to Figure 6.24, respectively.

As an illustration, the evolution of $\alpha$, $\beta$ and $\gamma$ parameters for one of the attributes of CRX database is shown in Figure 6.25(a), Figure 6.25(b) and Figure 6.25(c). It can be seen that these parameters mutate within the range and converge in the final stage in non-uniform mutation.

It can be seen that each database has different results with respect to the number of rules extracted and classification accuracy.

CRX database was selected for showing the classification accuracy when different minimum support values are used. The fuzzy rules were mined by using probabilistic
Chapter 6. Genetic Network Programming for Extracting and Applying Fuzzy Classification Rules

Figure 6.9: Number of rules in Waveform

Figure 6.10: Number of rules in CRX

Figure 6.11: Number of rules in Ionosphere

Figure 6.12: Number of rules in Heart

Figure 6.13: Number of rules in Pima

Figure 6.14: Number of rules in Pageblocks

Figure 6.15: Number of rules in Glass

Figure 6.16: Number of rules in Iris
6.4. Simulations

Figure 6.17: Accuracy in Waveform

Figure 6.18: Accuracy in CRX

Figure 6.19: Accuracy in Ionosphere

Figure 6.20: Accuracy in Heart

Figure 6.21: Accuracy in Pima

Figure 6.22: Accuracy in Pageblocks

Figure 6.23: Accuracy in Glass

Figure 6.24: Accuracy in Iris
Figure 6.25: Evolution of the fuzzy parameters by uniform and non-uniform mutation.

transitions in GNP and non-uniform mutation. As can be seen from Figure 6.26, the small support value works better in the proposed model, because it produces
more rules, which in turn provides more accurate classifier. Especially, increasing the minimum support value above a certain level leads to the decrease in the classification accuracy. The different databases show the same pattern.

![Figure 6.26: Classification accuracy for CRX DB for different minimum support values.](image)

In terms of the classification accuracy, the proposed model has been compared with seven associative classifiers described in the literature: CMAR [64], CBA [65], MCAR [66], ARC-PAN [67], CPAR [68], MSR [69], OAC [70], three very well known classification methods: C4.5, Support Vector Machine and Naive Bayes Classifier [71] and three algorithms based on evolutionary computing: CEFR [40], ESIA [72], BGP [73]. These methods were selected because they have obtained good results in comparison with other data mining classification systems and because they have been applied to some of the datasets used in our experiments. The comparison was performed under the same conditions in terms that: all of them deal with continuous values, they deal with classification problems and evaluate their models through the classification accuracy. The comparison was also performed under the same conditions with CEFR method due to the following reasons: CEFR is an evolutionary computation based method (Genetic Programming) that deals with continuous attributes using fuzzy discretization, discovers fuzzy classification rules based on a fuzzified database, uses three kinds of fuzzy membership functions (low, middle and high) and evolves the parameters of the fuzzy membership functions. Table 6.4, Table 6.6 and Table 6.5 show the classification accuracy of fourteen methods including the proposed one on the eight datasets listed in Table 6.2. If there is no available published results, the symbol "-" is placed. The last row of the tables shows the average performances of the methods for the eight datasets. Results using ten-fold cross validation are marked...
Chapter 6. Genetic Network Programming for Extracting and Applying Fuzzy Classification Rules

by the symbol "*".

Table 6.4: Classification accuracy (%) of GNP-Fuzzy algorithm and relevant association rule-based classification models.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>GNP-Fuzzy*</th>
<th>CMAR</th>
<th>CBA*</th>
<th>MCAR*</th>
<th>ARC-P*</th>
<th>CPAR</th>
<th>MSR*</th>
<th>OAC*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waveform</td>
<td>86.4</td>
<td>83.2</td>
<td>80.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CRX</td>
<td>88.9</td>
<td>84.9</td>
<td>84.7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>93.3</td>
<td>91.5</td>
<td>92.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Heart</td>
<td>88.2</td>
<td>82.2</td>
<td>82.2</td>
<td>81.1</td>
<td>83.5</td>
<td>83.7</td>
<td>81.4</td>
<td>81.1</td>
</tr>
<tr>
<td>Pima</td>
<td>79.6</td>
<td>75.1</td>
<td>72.9</td>
<td>78.5</td>
<td>75.0</td>
<td>72.6</td>
<td>73.3</td>
<td>78.1</td>
</tr>
<tr>
<td>Pageblocks</td>
<td>92.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Glass</td>
<td>78.8</td>
<td>70.1</td>
<td>73.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Iris</td>
<td>96.7</td>
<td>94.0</td>
<td>94.7</td>
<td>95.3</td>
<td>94.7</td>
<td>93.4</td>
<td>95.4</td>
<td>94.0</td>
</tr>
<tr>
<td>Average</td>
<td>88.0</td>
<td>83.0</td>
<td>82.9</td>
<td>84.9</td>
<td>84.4</td>
<td>86.1</td>
<td>83.3</td>
<td>84.4</td>
</tr>
</tbody>
</table>

Table 6.5: Classification accuracy (%) of GNP-Fuzzy algorithm and other evolutionary classification models.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>GNP-Fuzzy*</th>
<th>CERF*</th>
<th>ESIA*</th>
<th>BGP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waveform</td>
<td>86.4</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CRX</td>
<td>88.9</td>
<td>84.7</td>
<td>77.3</td>
<td>-</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>93.3</td>
<td>88.6</td>
<td>-</td>
<td>89.2</td>
</tr>
<tr>
<td>Heart</td>
<td>88.2</td>
<td>82.2</td>
<td>74.4</td>
<td>-</td>
</tr>
<tr>
<td>Pima</td>
<td>79.6</td>
<td>-</td>
<td>70.1</td>
<td>72.5</td>
</tr>
<tr>
<td>Pageblocks</td>
<td>92.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Glass</td>
<td>78.8</td>
<td>-</td>
<td>72.4</td>
<td>-</td>
</tr>
<tr>
<td>Iris</td>
<td>96.7</td>
<td>95.3</td>
<td>95.3</td>
<td>94.1</td>
</tr>
<tr>
<td>Average</td>
<td>88.0</td>
<td>87.7</td>
<td>77.9</td>
<td>85.2</td>
</tr>
</tbody>
</table>

It can be seen from Table 6.4 and Table 6.5 that the proposed method considerably outperforms all associative classifiers and evolutionary-based algorithms. Table 6.6 shows that the average classification accuracy of SVM and NBC are higher than GNP-Fuzzy, however they only consider two databases. Comparing the accuracy of each database, the proposed GNP-Fuzzy method clearly outperforms SVM and NBC classification models. Table 6.7 and Table 6.8 show the number of classification rules for GNP-Fuzzy algorithm and other algorithms for three datasets. The number of classification rules to achieve the accuracies was not published by the authors.
Table 6.6: Classification accuracy (%) of GNP-Fuzzy algorithm and other classification models.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>GNP-Fuzzy*</th>
<th>C4.5</th>
<th>SVM</th>
<th>NBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waveform</td>
<td>86.4</td>
<td>78.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CRX</td>
<td>88.9</td>
<td>84.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>93.3</td>
<td>90.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Heart</td>
<td>88.2</td>
<td>86.8</td>
<td>84.1</td>
<td>84.4</td>
</tr>
<tr>
<td>Pima</td>
<td>79.6</td>
<td>75.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Pageblocks</td>
<td>92.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Glass</td>
<td>78.8</td>
<td>68.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Iris</td>
<td>96.7</td>
<td>96.0</td>
<td>96.6</td>
<td>94.0</td>
</tr>
<tr>
<td>Average</td>
<td>88.0</td>
<td>82.8</td>
<td>90.3</td>
<td>89.2</td>
</tr>
</tbody>
</table>

in five from ten algorithms in Table 6.4. Considering the classifiers in Table 6.7, it can be seen that GNP-Fuzzy generates a large number of rules. However, the minimum support and confidence thresholds were not published by the authors in the compared algorithms. Table 6.8 shows the number of extracted rules obtained by the proposed method using different minimum support values. As expected, the number of interesting fuzzy rules obtained decreases as the minimum support value increases. In the proposed model, a small support value works better because it produces more rules, which in turn provides a more accurate classifier.

Table 6.7: Number of classification rules of GNP-Fuzzy algorithm with support value of 0.01 and other algorithms.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>GNP-Fuzzy*</th>
<th>ESIA*</th>
<th>CBA*</th>
<th>MCAR*</th>
<th>OAC*</th>
<th>ARC-PAN*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>115.0</td>
<td>64.0</td>
<td>5.0</td>
<td>31.0</td>
<td>9.0</td>
<td>60.0</td>
</tr>
<tr>
<td>Heart</td>
<td>16389.0</td>
<td>34.0</td>
<td>52.0</td>
<td>31.0</td>
<td>157.0</td>
<td>80.0</td>
</tr>
<tr>
<td>Pima</td>
<td>2885.0</td>
<td>36.0</td>
<td>45.0</td>
<td>66.0</td>
<td>112.0</td>
<td>50.0</td>
</tr>
<tr>
<td>Average</td>
<td>6463.0</td>
<td>44.6</td>
<td>34.0</td>
<td>42.6</td>
<td>92.6</td>
<td>63.3</td>
</tr>
</tbody>
</table>

6.5  Summary

In this chapter, a fuzzy rule-based classification method using an evolutionary algorithm has been proposed. It is shown that the GNP’s characteristics make the proposed model easy to formulate and use. Since the accuracy of a classification
Table 6.8: Number of classification rules of GNP-Fuzzy algorithm with different support values.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>0.01</th>
<th>0.05</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>115.0</td>
<td>93.0</td>
<td>62.0</td>
<td>34.0</td>
<td>27.0</td>
</tr>
<tr>
<td>Heart</td>
<td>16389.0</td>
<td>12561.0</td>
<td>4672.0</td>
<td>711.0</td>
<td>110.0</td>
</tr>
<tr>
<td>Pima</td>
<td>2885.0</td>
<td>471.0</td>
<td>135.0</td>
<td>28.0</td>
<td>9.0</td>
</tr>
<tr>
<td>Average</td>
<td>6463.0</td>
<td>4375.0</td>
<td>1623.0</td>
<td>257.6</td>
<td>48.6</td>
</tr>
</tbody>
</table>

model can be largely affected by the partitioning of continuous attributes, a fuzzy discretization technique is proposed for dealing with databases with continuous data. Fuzzy set theory allows the intervals to overlap instead of crisp sets, giving different membership values, not only 1 and 0 given by traditional discretization methods. Besides, the mined rules are expressed in linguistic terms, which are more natural and understandable for the users. Therefore, one of the main advantages of the proposed method over other techniques is the interpretability in terms of fuzzy if-then rules. Therefore, the main contribution on this work is described as follows:

- The fuzzy mining method is a tool for extracting a large number of fuzzy rules through generations and store them in a general pool. Using the GNP’s structure, there is no need to explicitly encode the fuzzy association rules into the genome of the GNP individual, instead, GNP individuals are used just to extract fuzzy rules through generations.

- The effectiveness of the proposed fuzzy discretization technique was clarified since the result is a set of fuzzy class association rules which will be used directly for the final classifier.

- The proposed algorithm is based on the evolution of fuzzy association rule sets, where the parameters of the fuzzy membership functions are also evolved becoming different rule by rule. As a result, more effective and stronger rules are extracted according to the progress of the evolution.

- The evolution of the fuzzy membership functions by non-uniform mutation shows the effectiveness for discovering new fuzzy rules generation by generation. This leads to the improvement of the classification accuracy.
• GNP-based fuzzy rule mining using non-uniform mutation extracts more rules than GNP with uniform mutation. This is because, non-uniform mutation makes use of searching the space uniformly at early generations and very locally at later generations. In other words, the non-uniform mutation has the advantage of higher probability of random search at early stages and much better local fine-tuning ability at later stages. This leads to discover new fuzzy rules generation by generation and the improvement of the classification accuracy. On the other hand, GNP with probabilistic transitions considerably extracts more rules than GNP with deterministic transitions. This is because GNP with probabilistic transitions has probabilistics natures, i.e., it has the possibility to produce many types of rules.

• The proposed method builds a classifier based on the degree of matching of the rules with data. As a result, it takes into account of the average distance between rules and data.

• Support, confidence and $\chi^2$ values are used to measure the significance of the rules, therefore, more effective rules are involved in the classifier model.

• In case of continuous datasets, one general problem with all associative classifiers is that they make use of discretization algorithms over the training dataset. The proposed method makes use of a fuzzy discretized dataset associated with continous values during the classification process.

• The other methods obtain lower classification accuracy, because it is generally difficult for them to deal with a large number of continous attributes in the dataset. For instance, the evolutionary algorithm described in paper [40], has the difficulty in coping with such a relatively good number of attributes being fuzzified.

• The achieved accuracy reveals the effectiveness of the proposed fuzzy class association rule mining and classification. These results are considered very promising, taking into account that the proposed model mines fuzzy rules which are more understable for the examiners.
Chapter 7

Concluding Remarks

A lot of different works have been done on association rule mining techniques using different methods and different approaches. While a significant number of theoretical investigations are expected to appear, an increasingly rich variety of applications and implementations will be motivated. In this thesis, a method of association rule mining using Genetic Network Programming with adaptation and self-adaptation mechanism was proposed. Furthermore, two models based on GNP for dealing with continuous transactions have been introduced. How the proposed fuzzy association rule mining method can be applied to classification problems has been also studied in this thesis.

7.1 Conclusions

The experimental results have shown the advantages and effectiveness of the proposed methods and classifier. More concretely, a deeper analyses of the obtained results are summarized as bellow:

- It was clarified that the adaptation and self-adaptation mechanisms improves the general performance of the conventional GNP-based data mining method, especially increasing the number of rules extracted and decreasing the number of generations where the extraction starts. Concretely speaking, the adaptation and self-adaptation mechanism adapt automatically the rates and proportion of the GNP individuals for crossover and mutation, determining which genetic operators contribute to the evolution of individuals most and which of them is most likely to find new rules.

- With reference to dealing with continuous attributes, it was confirmed that
Chapter 7. Concluding Remarks

GNP is able to extract a large number of rules without the bloating problem usually found in other evolutionary algorithms such as Genetic Programming. Using the GNP’s structure, there is no need to explicitly encode the association rules into the genome of the GNP individual, instead, GNP individuals are just used to extract association rules through generations. Therefore, the proposed GNP-based method is a tool for extracting a large number of rules through generations and store them in a general pool. Whereas, in other evolutionary algorithms the final result of the evolution process is the set of individuals.

- A correlation measure was employed to enhance the support-confidence framework, where the rule is measured not only by its support and confidence but also by a correlation measure. Among many different correlation measures, the \( \chi^2 \) test for independence is one of the most important because of its solid statistical basis. In the proposed methods, \( \chi^2 \) was used as a correlation measure that is easily calculated by the GNP’s structure.

- Taking the GNP’s structure into account, the extraction of association rules was done without identifying frequent itemsets as most Apriori-based data mining algorithms. The frequent item set searching would be the bottleneck of the association rule mining algorithms due to its long searching time.

- With reference to fuzzy association rules extraction, the proposed model is based on the evolution of fuzzy association rule sets, where the parameters of the fuzzy membership functions are also evolved and they become different rule by rule. As a result, more effective and stronger rules are extracted according to the progress of the evolution. This is a unique characteristic of the GNP-based approach because in most of the fuzzy mining techniques the parameters of the fuzzy membership functions are the same for all fuzzy rules extracted.

- It was proved that the proposed classifier considerably outperforms all the compared methods. The achieved accuracy reveals the effectiveness of the proposed fuzzy association rule mining and classification, taking into account the advantage to mine association rules that are expressed in linguistic terms, which are more natural and understandable for the users.
7.2 Future work

Recent studies on association rule-based classification have started to focus on the trade-off between interpretability and accuracy. This is because obtaining high degrees of interpretability and accuracy is a contradictory purpose, and, in practice, one of the two properties prevails over the other. Even though the proposed GNP-based fuzzy classifier has presented a promising results, the improvement of the interpretability is a central issue in the current research, where not only the accuracy receives attention but also the compaction and the interpretability of the obtained fuzzy rules. Thus, current work tries to minimize the complexity of fuzzy rule-based classifier while maximizing its accuracy by selecting only a small number of fuzzy rules from a large number of candidate rules. Whereas the definition of accuracy is straightforward, the definition of interpretability is rather problematic. Thus, the following issues have to be taken into account: (1) the number of rules is enough to be comprehensible and increase the accuracy, (2) rules premises should be easy in structure and contain only a few input variables, (3) linguistic terms should be intuitively comprehensible. Consequently, to realize an effective multi-objective genetic rule selection as a post-processing step for designing a compact, comprehensible and accurate GNP-based fuzzy associative classifier is the future work.

On the other hand, extending and improving the performance of the proposed models by using parallel processing will be considered in order to ensure the scalability of the proposed algorithms.
Acknowledgements

I would like to thank all people who have helped and inspired me during my doctoral study.

I specially want to express my deep and sincere gratitude to my professor, Dr. Kotaro Hirasawa. His wide knowledge, skill in many areas and assistance in writing reports has been of great value for me. His understanding and personal guidance have provided a great basis for the present thesis. I must also acknowledge Dr. Yoshie and Dr. Iwaihara for their evaluation and suggestions in this study.

Furthermore, I want to thank my family. The encouragement and support from my beloved husband Eloy and our always beautiful and joyful children Nicolas and Natalia is a powerful source of inspiration and energy. Without their love, understanding and encouragement it would have been impossible for me to finish this work. This thesis is specially dedicated to them.

A special thought is devoted to my parents Carlos and Ninfa for a never-ending support. In particular, I do not have words to thanks my mother Ninfa. I remember many sleepless nights with her accompanying me and her constant support when I encountered difficulties. My special gratitude is due to my sister Melvy for her love and support.

My loving thanks are due to Mr Fujio, Yasuko and Masayuki Tawara, Mr Soichi Fukuoka and Ms Naoko Nishino. They let me own a happy stay in Japan. It was through their kindness, help and consideration that I completed this study, I owe them my eternal gratitude.

This research would not have been possible without the financial assistance of the Japan Society for the Promotion of Science and Waseda University IPS, I express my deep gratitude to those institutions.

Finally and the most important, thanks to God for my life through all tests in the past five years. You have made my life more beautiful. Be Your name exalted, honored, and glorified.
List of Publications

Journals


• **J6**  E. Gonzales, K. Taboada, K. Shimada, S. Mabu and K. Hirasawa, “Combination of Two Evolutionary Methods for Mining Association Rules in Large

**International Conferences**

- **C1**  

- **C2**  

- **C3**  

- **C4**  

- **C5**  

- **C6**  


References


References


References


the Congress on Evolutionary Computation, La Jolla, CA, USA, pp. 633-638, 2000.