

2009-01-01

Identification of Human Gait Using Genetic Algorithm Tuned Fuzzy Logic

Abdallah Abdel-Rahman Hassan Mahmoud
University of Texas at El Paso, aahassan@miners.utep.edu

Follow this and additional works at: https://digitalcommons.utep.edu/open_etd



Part of the [Biomedical Commons](#), and the [Computer Sciences Commons](#)

Recommended Citation

Hassan Mahmoud, Abdallah Abdel-Rahman, "Identification of Human Gait Using Genetic Algorithm Tuned Fuzzy Logic" (2009).
Open Access Theses & Dissertations. 278.
https://digitalcommons.utep.edu/open_etd/278

This is brought to you for free and open access by DigitalCommons@UTEP. It has been accepted for inclusion in Open Access Theses & Dissertations by an authorized administrator of DigitalCommons@UTEP. For more information, please contact lweber@utep.edu.

IDENTIFICATION OF HUMAN GAIT USING GENETIC ALGORITHM
TUNED FUZZY LOGIC

Abdallah Abdel-Rahman Hassan Mahmoud
Department of Electrical and Computer Engineering

APPROVED:

Thompson Sarkodie-Gyan, Ph.D., Chair

Scott Starks, Ph.D.

Bill Tseng, Ph.D.

Jody Riskowski, Ph.D.

Patricia D. Witherspoon, Ph.D.
Dean of the Graduate School

For my parents, my wife, and my daughter

IDENTIFICATION OF HUMAN GAIT USING GENETIC ALGORITHMS
TUNED FUZZY LOGIC

By

Abdallah Abdel-Rahman Hassan Mahmoud, B.Sc.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

THE UNIVERSITY OF TEXAS AT EL PASO

May 2009

Acknowledgements

I would like to express my appreciation to the many people who have helped me in completing this thesis. Specifically, I thank Dr. Thompson Sarkodie-Gyan for his support, patience, and guidance during these past two years. I gratefully thank Mrs. Huying Yu, Mr. Chad Macdonald, and Mr. Murad Alaqtash for their kind permission to use the data they have collected, tracked and processed. I would also like to thank all the subjects from the campus of the University of Texas at El Paso who participated in this study.

The completion of this thesis was possible because of the patience and understanding from my wife. In particular, I would like to express my complete thanks and gratitude to my parents for their support since I came to this world.

Abstract

Data mining is concerned with the discovery of useful hidden information in large databases. Classification is a data mining task producing rules in which a set of attributes in data predict the value of a class attribute. Classifiers usually produce a large number of rules, most of which are not interesting to the user. Rule interestingness is a decisive factor. However, evaluating rule interestingness is challenging as it involves both objective (data-driven) and subjective (user-driven) aspects.

In this research, a fuzzy genetic algorithm is proposed to discover classification rules that are both accurate and interesting. Continuous attributes are fuzzified so that the produced rules are fuzzy rules stated in terms that are more natural to users and easier to measure. A weighted fitness function is used with two elements: the first is an objective interestingness measure based on the attribute information gain, and the second is a predictive accuracy measure.

Classification of human gait dynamics data is useful for rehabilitation processes. Three variants of the proposed fuzzy genetic algorithm are experimented for different classification tasks performed on a collected gait dynamics dataset from 23 participating healthy subjects. Part of the dataset is used for training of the genetic algorithm and the other part is used to test the performance of the genetic algorithm. Promising results are obtained specially for identifying the human subjects based on their gait dynamics, and mapping an unknown subject to a previously known subject with similar gait parameters.

Table of Contents

Acknowledgements	iv
Abstract	v
List of Tables	viii
List of Figures	ix
Chapter 1. Introduction	1
1.1 Motivation	2
1.2 Objectives	3
1.3 Organization of the Thesis	4
Chapter 2. Background and Significance	5
2.1 Data Mining and Knowledge Discovery in Databases KDD	6
2.1.1 Data Mining	6
2.1.2 Rule Interestingness	10
2.2 Soft Computing	11
2.2.1 Fuzzy Logic	11
2.2.2 Genetic Algorithms	12
Chapter 3. Design of the Proposed Fuzzy Genetic Algorithm	16
3.1 Introduction	17
3.2 Objectives of the Proposed Algorithm.....	19
3.3 Genetic Encoding	21
3.4 Rule Fitness Evaluation	24

3.4.1 Predictive Accuracy Measure	29
3.4.2 Interestingness Measure	32
Chapter 4. Classification Experiments	38
4.1 Gait Data Collection	39
4.2 Fuzzification of Continuous Attributes	48
4.3 Gait Identification Experiments	54
4.4 Gait Similarity Experiments	59
4.5 Gender Classification Experiments	62
4.6 BMI Classification Experiments	65
4.7 Speed Classification Experiments	68
4.8 Age Classification Experiments	71
Chapter 5. Results and Interpretations	74
5.1 Results for Gait Identification Experiments	75
5.2 Results for Gait Similarity Experiments	82
5.3 Results for Gait Classification Experiments	89
Chapter 6. Conclusions and Future Considerations	91
6.1 Conclusions	92
6.2 Future Considerations	95
References	96
Curriculum vitae	100

List of Tables

Table 2.1 Rule interestingness measures in literature	10
Table 3.1 Sample weather dataset	26
Table 4.1 Subjects participating in gait data collection for this thesis	40
Table 4.2 Fuzzy values of attributes of all subjects	60
Table 4.3 Similarity matrix between training (rows) and testing (columns) subjects	61
Table 4.4 Properties of subjects in training and test datasets	62
Table 5.1 Results for the GA variant with accuracy only (Gait Identification)	75
Table 5.2 Results for the GA variant with fixed weights (Gait Identification)	76
Table 5.3 Results for the GA variant with adapting weights (Gait Identification)	77
Table 5.4 Results for combining fixed and adapting weights (Gait Identification)	78
Table 5.5 Results for gait similarity experiments (Fixed Weights)	85
Table 5.6 Results for gait similarity experiments (Adapting Weights)	86
Table 5.7 Results for gait similarity experiments (Combined Fixed/Adapting Weights)	87
Table 5.8 Summary of results for gait similarity experiments (all experiments)	88
Table 5.9 Summary of results for gait similarity experiments grouped by speed	88
Table 5.10 Results for the GA variant with fixed weights (Gender Classification)	89
Table 5.11 Results for the GA variant with adapting weights (Gender Classification)	89
Table 5.12 Results for combining fixed and adapting weights (Gender Classification)	90

List of Figures

Figure 3.1 Encoding of candidate rules	22
Figure 3.2 An example of attribute absence flag	22
Figure 3.3 A trapezoidal fuzzy mapping function for the temperature attribute	27
Figure 3.4 A trapezoidal fuzzy mapping function for the humidity attribute	29
Figure 4.1 Specialized treadmill used for collecting ground reaction forces	41
Figure 4.2 The z-component of the force on the left plate for three strides	43
Figure 4.3 Normalized EMG activity of soleus muscle during one gait cycle	44
Figure 4.4 Normalized EMG activity of tibialis anterior muscle during one gait cycle	45
Figure 4.5 Normalized EMG activity of gastrocnemius muscle during one gait cycle	46
Figure 4.6 Normalized EMG activity of vastus lateralis muscle during one gait cycle	47
Figure 4.7 Stride time fuzzy membership function	48
Figure 4.8 Stride length fuzzy membership function	49
Figure 4.9 Stride width fuzzy membership function	49
Figure 4.10 Stance/Swing fuzzy membership function	49
Figure 4.11 Doble Stance ratio fuzzy membership function	50
Figure 4.12 Left force plate feature fuzzy membership function	50
Figure 4.13 Soleus peak shift fuzzy membership function	50
Figure 4.14 Tibialis anterior peak shift fuzzy membership function	51
Figure 4.15 Gastrocnemius peak shift fuzzy membership function	51
Figure 4.16 Vastus lateralis peak shift fuzzy membership function	51

Figure 4.17 Body mass index fuzzy membership function	52
Figure 4.18 Height fuzzy membership function	52
Figure 4.19 Age fuzzy membership function	52
Figure 4.20 Adapting weights for the genetic algorithm	55
Figure 4.21 Identifying the subject of a stride using rule sets of all subjects	57
Figure 4.22 Identifying the subject based on an experiment	58
Figure 4.23 classifying the gender of a subject based on an input stride	63
Figure 4.24 classifying the gender of a subject based on an experiment	64
Figure 4.25 classifying the BMI of a subject based on an input stride	66
Figure 4.26 classifying the BMI of a subject based on an experiment	67
Figure 4.27 classifying the speed of a subject based on an input stride	69
Figure 4.28 classifying the speed of a subject based on an experiment	70
Figure 4.29 classifying the age group of a subject based on an input stride	72
Figure 4.30 classifying the age group of a subject based on an experiment	73

CHAPTER 1

Introduction

1-1 Motivation

The amount of data stored in data bases continues to rapidly grow [1]. For example, the amount of electronically stored information in the United States alone was estimated in the year 2002 to range between 450 and 750 petabytes [2], equivalent of up to 375 trillion document pages and records [2]. Wal-Mart uploads 20 million point-of-sale transactions every day to a massively parallel system with 483 processors running a centralized database [3]. However, raw data by itself provides limited knowledge. The current state is what is called “Data rich, Information poor” state [3]. The need for tools to discover the valuable knowledge hidden in massive datasets is crucial. The main goal of data mining research is to provide such tools.

Data mining still faces many challenges that limit its wide use to some extent. Traditional data mining techniques usually produce a large number of rules to the user that is difficult (or maybe impossible) for a human to evaluate. The fact is that most of these rules that are supposed to be the discovered knowledge are either just noise in the data or maybe pieces of knowledge that were obviously known to the users. For that reason, users usually underestimate the power and need for data mining and knowledge discovery techniques. Moreover, the form of the resultant rules might not be comprehensible to humans, for example when using neural networks for data mining, the way knowledge is hidden in a trained neural network is very difficult for humans to understand [4].

For the previous reasons, it can be concluded that the main challenge for techniques trying to discover hidden knowledge in a data base is to be able to present to users what might be really interesting to them and easy to understand. This is a difficult task involving different criteria, evaluating rule interestingness has an objective element (driven solely from data), and a subjective element (domain dependent; needs a human expert). Unfortunately, the majority of rule induction literature focuses on discovering accurate rather than interesting rules [5].

The application that was chosen for this thesis is the human gait analysis, with large and increasing percentage of people suffering from gait impairments; it is highly desired to discover hidden cause and effect relationships among the large number of attributes that govern the human gait. Many of these attributes are difficult to measure and it is still difficult to say why a specific patient is not able to walk easily as a healthy subject.

1-2 Objectives

The main objective of this thesis is apply different variants of a fuzzy genetic algorithm to try to discover classification rules (hidden knowledge) in a dataset of human gait collected from healthy subjects. If these rules can be discovered, better understanding of the relationships between different entities and attributes of gait can be reached, and this knowledge can then be applied for rehabilitation processes for patients.

The main reason of studying genetic algorithms for this problem is the vast search space for the human gait analysis specially as data is collected from more healthy and impaired subjects. The number of attributes to be considered is large (kinetic, kinematic, EMG, etc.).

1-3 Organization of the thesis

This thesis is composed of six chapters. Chapter 2 provides background knowledge about the fields involved in this study; data mining, genetic algorithms, fuzzy logic, and rule interestingness measures. Chapter 3 discusses in detail the design of the fuzzy genetic algorithm used for the experiments of this study. Chapter 4 discusses how gait dataset for the experiments was collected, what data attributes were used, and the different classification experiments that were run. Chapter 5 shows the results of running the various classification experiments and the interpretation of these results. Chapter 6 provides the conclusions of this study and considerations for future work.

CHAPTER 2

Background and Significance

2-1 Data Mining and Knowledge Discovery in Databases KDD

2-1-1 Data Mining

Nowadays data accumulates at an alarming speed in various storage devices, and so does valuable information. According to Ntoutsis et al [6], the world produces between 1 and 2 exabytes of unique information per year, which is roughly 250 megabytes for every man, woman, and child on earth. However it is difficult to understand information hidden in data without the aid of data analysis techniques, which has provoked extensive interest in developing a field separate from machine learning which is data mining [4].

Data mining has successfully provided solutions for finding information from data in bioinformatics, pharmaceuticals, banking, retail, sports, and entertainment, etc. It has been one of the fastest growing fields in computer industry. It is used with great success in the business world in areas such as marketing [7].

There are abundant definitions of data mining in the literature ranging from simple brief definitions to very detailed ones specifying data mining methodologies, objectives, and other characteristics.

One simple definition by Dunham [8] states that "*data mining is often defined as finding hidden information in a database*". This definition concentrates only on the most important aspect of data mining - searching for hidden information, with no hint concerning the techniques

used, the form or properties of the hidden data to be found, or even the necessity of finding such data. A similar definition is given by Padmanabhan and Tuzhilin [9] who state that "*Data mining is the process of discovering hidden structure or patterns in data*".

Marakas [10] defines data mining as the "*set of activities used to find new, hidden, or unexpected patterns in data*". Though this definition is still a very brief one, it adds an important issue to the previous definitions. This definition states that output information of data mining should be new or unexpected. There might be patterns in data that are highly accurate, but data mining is not concerned with such patterns if they are previously known or expected by user.

The properties of information discovered by data mining is more described by Hilderman and Hamilton [11] who define data mining as "*the efficient discovery of previously unknown, valid, novel, potentially useful, and understandable patterns in large databases*". In addition to being new and unexpected as mentioned in Marakas' definition, discovered patterns need to be useful and understandable to user. The validity property stated in this definition might not be considered an addition to previous definitions as validity of patterns can be considered implied in all previous definitions. An important aspect of this definition is that it confines data mining to large databases, which is plausible practically. In real life, data mining is usually applied to large datasets, not to small size databases.

Usefulness and Unexpectedness of discovered patterns are also mentioned as main goals of data mining in the definition of Giannella and Robertson [12] which states that data mining is "*the analysis of (often large) observational data sets to find unsuspected relationships and to*

summarize the data in novel ways that are both understandable and useful to the data owner".

Here also data mining is stated to be more applicable to large data sets rather than small ones.

Anderson [2] defines data mining as "*a computerized technology that uses complicated algorithms to find relationships and trends in large databases, real or perceived, previously unknown, to promote decision support*". According to this definition, the objective of research in data mining should be to improve the decision-making process of an organization, but describing the algorithms used in data mining systems as "*complicated*" might not be accurate as some data mining algorithms might be simple and efficient.

Freitas [1] provides another definition for data mining that is more specific in stating the methods used by data mining algorithms, he defines data mining as "*an interdisciplinary field, using methods of several research areas, specially machine learning and statistics, to extract high-level knowledge from real-world data sets*". Another important aspect of this definition is characterizing the extracted knowledge to be high-level; the forms of this high-level knowledge depend on the data mining task as described in more detail in the next section. The data set operated upon by data mining is described to be real-world, practically speaking, data mining should be able to handle problems in real-world datasets like missing data, inaccurate data, and massive size.

A definition of data mining that tries to combine the important factors mentioned in the previous definitions can be stated as follows: *Data mining is the computerized application of machine learning and statistical techniques to discover new or unexpected, useful, and*

understandable knowledge in large real-world databases to improve the decision making process.

Finally, the term "**Knowledge Discovery in Databases**" **KDD** is used by some references like [14] as a synonym of data mining. However, other references differentiate between the two terms [8, 10] where knowledge discovery in databases is used to refer to a process consisting of many steps, while data mining is only one of these steps. KDD may include other processes like data preprocessing, transformation, and interpretation [8]. Data collection and storage are sometimes not included in data mining [4] though they are important for data mining. Redundant or irrelevant information exists in datasets, and inconsistent formats of collected datasets may disturb the processes of data mining, even mislead search directions, and degrade results of data mining. This happens because data collectors and data miners are usually not from the same group, i.e. in most cases data are not originally prepared for the purpose of data mining [4].

The data mining tasks can be divided into two broad categories [8]; predictive tasks and descriptive tasks. Although the term “data mining” is a new term proposed in recent decades, the tasks of data mining, such as classification and clustering, have existed for a much longer time [4]. With the objective to discover unknown patterns from data, methodologies of data mining are derived from machine learning, artificial intelligence, and statistics, etc. Data mining techniques have begun to serve fields outside of computer science and artificial intelligence, such as the business world and factory assembly lines. The capability of data mining has been proven

in improving marketing campaigns, detecting fraud, predicting diseases based on medical records, etc [4].

2-1-2 Rule Interestingness

In literature, various approaches have been applied to calculate a numerical estimate for rule interestingness [14]. Most of these methods are objective, which means they statistically driven from the dataset, and this will be our approach in this study, while the rest of them are subjective methods, meaning they are user-driven estimates. Table 2.1 summarizes some of these methods. For a detailed discussion of them, refer to reference [14].

Table 2.1 – Rule interestingness measures in literature

Interestingness measure	Foundation	Scope	Class
Piatetsky-Shapiro's rule interest	Probabilistic	Single rule	Objective
Smyth and Goodman's J measure	Probabilistic	Single rule	Objective
Major and Mangano's refinement	Probabilistic	Single rule	Objective
Kamber & Shighal interestingness	Probabilistic	Single rule	Objective
Liu et al. general impressions	Syntactic	Single rule	Subjective
Gago and Binto's Distance metric	Distance	Rule set	Objective
Freitas Surprisingness	Probabilistic	Rule set	Objective

2-2 Soft Computing

Soft computing is a general term covering a number of methodologies including fuzzy logic, neural networks, and evolutionary algorithms (genetic algorithms). It is hard to come up with a neat definition of the meaning and the boundaries of the disciplines involved [15]. Soft computing provides algorithms that are able to value, to reason, and to discriminate, rather than just to calculate [16]. The common thread through all the soft computing methodologies is that, unlike conventional algorithms, they are tolerant of imprecision, uncertainty, and partial truth [15]. These new structures of calculation are based on logic with more values (not just true or false, but different degrees of certainty) and inspired by natural processes like selection, aggregation, and cooperation [16].

2-2-1 Fuzzy Logic

Fuzzy set theory provides a mathematical framework for representing and treating uncertainty in the sense of vagueness, imprecision, lack of information, and partial truth [15].

Fuzzy logic is based on the premise that the key elements in the activity of human thinking are not numbers but rather indicators of fuzzy sets. A fuzzy set is a class of objects in which the transition from membership and non-membership to the class is gradual. In this way, every element belongs to a set with a determinate degree of membership [16]. The same element may belong to more than one fuzzy set with different degrees of membership. The concept of the

fuzzy set allows the representation of the imprecision of a given concept by means of the degree of the membership function.

The choice of the membership functions for each fuzzy set is wholly subjective and closely connected to the application. In many cases, the choice of the membership function can represent a problem and it is preferred to fall back on the indications of some expert in the field in question, or to optimization techniques [16].

2-2-2 Genetic Algorithms

Genetic algorithms have proven to be a valid procedure for global optimization, applicable in many sectors of engineering [16]. The basic genetic algorithm is simple but powerful in solving problems in various areas [4].

The technique of the genetic algorithms is based on the natural evolution principle. Among a population of points belonging to the domain of the optimization problem, the individual (candidate solution) who best adapts will be able to survive and reproduce, adapting here means to be evaluated to be close to the optimal solution of the problem to be solved.

A genetic algorithm (GA) does not process a single point in the solution domain at a time, but processes a population of points in parallel. The main advantage of that is that the possibility that the GA be trapped in a local maximum will be less than other techniques that handle one solution point at a time.

A GA run starts by randomly generating the set of points to start with (generation 1), then a set of GA operators can be applied to enhance the population of points (to let the population points come closer to the global maximum). At the last generation, the GA will be successful if one or more points in the population are close enough to the global maximum. Between the first and last generations, the applied GA operators might include:

- Reproduction (also called selection): a candidate solution point is copied as is from a generation to the next generation.
- Crossover (also called recombination): each point in the solution space is called a chromosome which is encoded as a set of elements called genes, the crossover is done by exchanging parts of two chromosomes in a generation to create two new chromosomes in the following generation. It can be said that the two chromosomes get married, and their two kids are expected to inherit the best of them and might be closer to the global solution.
- Mutation: a single bit or gene in a chromosome changes its value, this operator may help a chromosome that is far from the global maximum or that is trapped in a local maximum to move to another portion of the solution domain that can let it adapt to the required optimum point.

The three mentioned operators are usually executed with probabilities that can be tuned to achieve the results required from the GA. During the GA run the chromosomes are evaluated and chosen as inputs to the GA operators based on a fitness function. This should be carefully chosen such that for a candidate solution point with a higher fitness function, this is an indication that it is closer to the global maximum that the goal of the GA run.

To terminate the GA run, different stop criteria can be run. Usually the stop criterion is when the number of generations run so far reaches a certain number. Another technique that is also common is to allow a variable number of generations but to monitor the average or maximum fitness in a generation and to stop the GA run when this fitness value exceeds a certain threshold value.

One way of improving the performance of a GA is called elitism. This technique works by maintaining the population point with highest fitness value from a generation to the next generation by reproducing it.

GAs are specially useful under the following circumstances [4, 17]:

- The problem space is large, complex.
- Prior knowledge is scarce.
- It is difficult to determine a machine learning model to solve the problem due to complexities in constraints and objectives.
- Traditional search methods perform badly.

Genetic algorithms are known to be easily parallelized, much recent work has concentrated on the implementation of GAs on both fine and coarse grained parallel machines. This holds promise of significant decreases in the execution time of GAs [18].

The following steps [4] are a summary of the operations of a simple GA:

1. Figure out a way to encode solutions of the problem according to domain knowledge and required solution quality. This is not a trivial task and quite often it involves some additional heuristics [19].
2. Randomly generate an initial population of chromosomes which corresponds to solutions of the problem.
3. Calculate the fitness of each chromosome in the population pool.
4. Select two parental chromosomes from the population pool to produce offspring by crossover and mutation operators.
5. Go to step 3, and iterate until the stop criterion is met.

CHAPTER 3

Design of the Proposed Fuzzy Genetic Algorithm

3-1. Introduction

The goal of the data mining classification task is to map data into predefined groups or classes. When applied to a database, the classification rules will have the form [20]:

If some conditions on the values of predicting attributes are true

Then predict a value for the goal attribute

The antecedent part (if part) can be viewed as a conjunction of conditions of the form *<attribute operation value>* and the consequent (then part) is of the form *<goal = value>* where the number of classes equals the number of possible values of the goal attribute.

In a large database, the search space for the classification rules is huge due to the large number of attributes and their values. Accordingly, applying a genetic algorithm to search for accurate and interesting rules is very practical.

Another motivation for using a genetic algorithm in data mining emerges from the tendency of practical database systems to show rapid growth in size. This necessitates that the data processing systems be able to take care of their scalability so that their performance is not degraded when applied to large databases. Genetic algorithms are known to be easily parallelized, and so they can be assumed to be able to build scalable systems since parallelization is one major technique to achieve scalability.

Another reason for using of genetic algorithms in data mining tasks is mentioned by Noda et al [20]. He considers genetic algorithms to be less sensitive to attribute interaction problems than most rule induction methods. This is due to the method of operation of genetic algorithms where the fitness function evaluates a candidate rule as a whole so all interactions among attributes are taken into account. In contrast, most rule induction methods select one attribute at a time and evaluate a partially constructed candidate rule, rather than a full candidate rule.

When using a genetic algorithm for the data mining classification task, two approaches can be followed. First, a single large GA run can be held to discover rules for all the values of the goal attribute. Second, a number of separate GA runs can be held where each run tries to discover rules predicting a distinct value of the goal attribute, so the number of runs equals the number of possible values of the goal attribute. This work experiments the second approach so that each run tries to discover the best rules for a specific goal value.

The algorithm used here is based on the work of Noda et al [20] where the important aspect was to discover interesting prediction rules using a genetic algorithm. The work of Noda handled only datasets whose attributes are all nominal. The algorithm presented in this chapter extends the work of Noda in two aspects. First, it provides a method of dealing with continuous attributes as well as nominal ones. Each continuous attribute is fuzzified to a number of relevant fuzzy sets. This way of fuzzifying continuous attributes is similar to the technique used by Romao et al [21].

The second aspect in which our algorithm extends the work of Noda is the attempt to achieve equal performance with simpler genetic operators. In spite of using specific operators for insertion and removal of attributes from the candidate rules, we provide an alternative scheme by depending on modifying the initialization and mutation operators to handle attribute insertion and removal.

3-2. Objectives of the Proposed Algorithm

The work presented in this chapter has the following objectives:

- **Discovering rules that are both accurate and interesting:** It is required to find effective ways to discover rules that are both interesting and accurate using a genetic algorithm. The fitness function of the genetic algorithm is to evaluate both accuracy and interestingness for every candidate rule during the evolution of the genetic algorithm. Unlike the alternative approach which directs the fitness function to discover accurate rules and postpones the evaluation of their interestingness to a postprocessing step, our approach evaluates both accuracy and interestingness of rules concurrently. We believe that separating the evaluation of accuracy and interestingness results in decreasing the chance of appearance of interesting rules in the resultant rules generated by the genetic algorithm. To justify this assumption, consider three candidate rules representing the three possible cases of rules resulting from applying accuracy measure alone; *rule A*, *rule B*, and *rule C*, where: *rule A* is accurate but previously known to the user so it is not interesting, *rule B* seems to be accurate but represents noise in the data and it is

meaningless to user, while **rule C** represents an accurate rule that is new or unexpected to the user so it is interesting. If we only consider accuracy in the fitness function, rules like A and B will have fitness values comparable to rules like C which is the required ones. Rules of type C might even be hidden if the accuracy measure favors rules like A and B so in the postprocessing phase, the interestingness measure will have poor candidates to evaluate. If rules of type C are strengthened from the beginning of the processing by an interestingness measure, they will be favored to rules like A and B which are not required. This means a better chance to have interesting rules at the end of the classification task.

- **Dealing with both types of attributes; nominal and continuous:** Since practical databases include both types of attributes, it is required to build a system that is equally efficient with continuous and nominal attributes. It is required that the genetic encoding of both types of attributes take place in a similar fashion so that the genetic operators like crossover and mutation can behave similarly on both continuous and nominal attributes. Fuzzification of continuous attributes can help in achieving the above mentioned goal, since a fuzzified attribute can be viewed as a nominal attribute with each of its fuzzy sets corresponding to one nominal value. For example, if attribute **att1** is fuzzified to the three fuzzy sets: **low**, **medium**, and **high**, it will be encoded in the candidate rules as if it were a nominal attribute with three values. This way, both types of attributes will be encoded similarly and the effect of genetic operators on them will be similar.

- Producing rules that are easily understood by users. One of the main challenges of data mining systems is to produce truly comprehensible rules, using simple structure and terms familiar to the user. We believe that fuzzy terms are more natural to users than crisp operators.

3-3. Genetic Encoding

Each chromosome represents a candidate classification rule of the form:

If $att_1=val_1, att_2=val_2, \dots, att_n=val_n$ then class= $goal_i$

where n is the number of predicting attributes. If att_i is a nominal attribute, then val_i is one of the nominal values defined in the domain of att_i . If att_i is a continuous attribute, then val_i is one of the fuzzy sets defined by the user for att_i , so the rules represented by the chromosomes are fuzzy rules.

The consequent of the rule, the predicted class value, is not encoded in the chromosome. It is fixed for every run of the genetic algorithm. We evolve a separate run of the genetic algorithm for every value of the goal attribute. The number of goal values for practical datasets is usually small, so running a separate run for every goal value is applicable.

Physically, the chromosomes are fixed in length, with every chromosome composed of n genes, one for every predicting attribute as shown in the following figure.

val ₁	val ₂	val ₃	val _n
gene ₁	gene ₂	gene ₃		gene _n

Figure 3.1 – Encoding of candidate rules

In practice, the number of predicting attributes might be large. If every attribute appears in the antecedent of every candidate rule, the rule will be difficult for the user to make use of. Actually, understandable rules are short rules where the attributes appearing in the antecedent are those that actually affect the goal attribute. The encoding should be flexible to allow the number of predicting attributes to be variable. To achieve this without changing the simple physical representation shown in figure 3.1, a flag value of **-1** is used to indicate that a certain attribute is absent from a candidate rule. The following figure shows an example of an encoded rule to clarify how the flag value is used:

3	2	5	-1	0	-1
att ₁ present	att ₂ present	att ₃ present	att ₄ absent	att ₅ present	att ₆ absent

Figure 3.2 – An example of attribute absence flag

Although the rule encoded in figure 3.2 is represented by a chromosome composed of six genes representing six predicting attributes, only four attributes are present in the encoded rule; a gene value of -1 means that the predicting attribute corresponding to the gene is absent from this rule, att₄ and att₆ in figure 3.2. The antecedent of the rule represented by figure 3.2 is as follows:

If att₁=3 & att₂=2 & att₃=5 & att₅=0 then

Each gene can only hold an integer value. If the gene corresponds to a nominal predicting attribute, the value held in the gene will be the index of a nominal value from the domain of the attribute. For example, if we have a nominal attribute called **color** whose domain contains five values: **red, green, blue, white, black**. The index of the first color, **red**, is 0, the index of the second color, **green**, is 1 and so on. So if the gene of the attribute **color** holds the value of 3, it means that "**color=white**" is an element of the rule antecedent of the chromosome containing this gene.

In the case of a continuous predicting attribute, the gene corresponding to the attribute can also carry only an integer value. The value carried by the gene represents the index of a fuzzy set from the fuzzy sets defined for the attribute. For example, if we have a continuous attribute called **temperature** for which we define the three fuzzy sets: **cold, moderate, and hot**. The index of the first set, **cold**, will be 0, the index of the second set, **moderate**, will be 1, and the index of the last set, **hot**, will be 2.

Unifying the data type of the genes to the integer data type has the benefit of ensuring that all genes are operated on by the genetic operators in a similar fashion, so the behavior of the algorithm with continuous attributes does not differ from its behavior with nominal ones.

3-4. Rule Fitness Evaluation

The fitness function used for the proposed genetic algorithm has two elements; the first element represents a measure of the accuracy of the candidate rule, while the other element is an objective measure of the candidate rule interestingness. The two elements are normalized so that the values of both measures lie between zero and one.

The two normalized measures of accuracy and interestingness are combined together to give the fitness of a candidate rule using the following weighted function [13] which keeps the fitness normalized

$$fitness = \frac{w1 * Interestingness + w2 * Accuracy}{w1 + w2} \quad 3-1$$

As shown in equation 3-1, the weight of the accuracy measure is twice the weight of the interestingness measure. These weights can be adjusted easily relative to each other whenever it is required. In the following two subsections, the two measures are discussed in more detail. For the purpose of the following discussion, we will use a small sample dataset to clarify how accuracy and interestingness are calculated, the sample **weather** dataset, shown in table 3.1 below, is composed of 14 records and 6 attributes. This dataset is a modified version of the one used by Witten & Frank [22] where we added the first column **row #** to be able to refer to specific rows in the following discussion. We also converted the two attributes **temperature** and

humidity to be continuous attributes to clarify how fuzzification of continuous attributes fits in our proposed system.

The **weather** dataset shown in table 3.1 stores conditions under which some hypothetical outdoor game may be played. There are four predicting attributes; two of which are nominal, **outlook** and **windy**, and two predicting attributes are continuous, **temperature** and **humidity**. The class attribute is the attribute **play** with 2 classes; **yes** and **no**.

As discussed earlier, encoding of genes corresponding to nominal attributes in the chromosomes of candidate classification rules makes use of integer indexes of values in the domains of the nominal attributes, in our example, the nominal attribute **windy** has two possible values: **false** and **true**, the value **false** is of index 0 and the value **true** is of index 1. Similarly, the nominal attribute **outlook** has three possible values: **sunny**, **rainy**, and **overcast** that can be indexed with the integer values 0, 1, and 2 respectively.

As for the continuous attributes, a number of fuzzy sets is to be defined for each continuous attribute and a fuzzy function is to be defined for the boundaries of these sets. Our system uses the standard trapezoidal fuzzy function to map a crisp continuous value from the table to one or two of the fuzzy sets defined for the attribute. To clarify how this takes place, consider the continuous attribute **temperature** in table 3.1, we might define the three fuzzy sets **cold**, **mild**, and **hot**. Fig. 3.3 shows a possible trapezoidal fuzzy function for this fuzzified attribute.

Table 3.1 – Sample weather dataset

Row #	Outlook	Temperature (°C)	Humidity(%)	Windy	Play?
1	sunny	25	80	false	no
2	sunny	21	75	true	no
3	overcast	28	83	false	yes
4	rainy	17	85	false	yes
5	rainy	8	55	false	yes
6	rainy	5	65	true	no
7	overcast	10	45	true	yes
8	sunny	15	80	false	no
9	sunny	7	37	false	yes
10	rainy	13	42	false	yes
11	sunny	14	40	true	yes
12	overcast	18	90	true	yes
13	overcast	22	57	false	yes
14	rainy	14	73	true	no

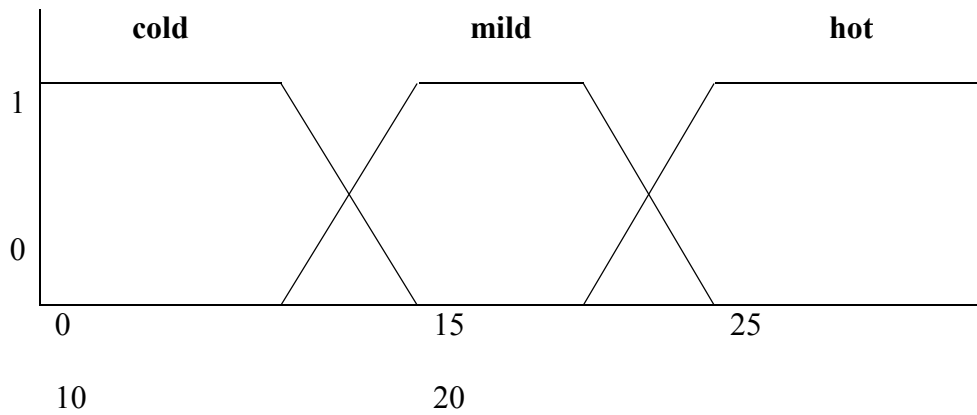


Figure 3.3 – A trapezoidal fuzzy mapping function for the temperature attribute

Now if we take the value of **temperature** attribute for any record of our sample **weather** dataset, we can map each crisp value for this continuous attribute to one or two of the three fuzzy sets in figure 3.3 with membership values ranging between 0 and 1. For example, the first record has the value of 25 for the **temperature** which is mapped to **hot** fuzzy set with membership value of 1, meaning that this value is a member only in this set, i.e. completely hot.

The second record in the table has the crisp value of 21 for the temperature, it is clear from figure 3.3 that this value lies in between the **mild** and the **hot** fuzzy sets, meaning that it will be mapped to both sets, we can see that this value is more **mild** than **hot** since it is nearer to the boundary of the **mild** set so its membership to **mild** set is larger the membership value to the **hot** fuzzy set. The two membership values will sum up to the value of 1 in the standard trapezoidal fuzzy mapping function. We can calculate the degree of membership to the hot fuzzy set using the following equation

$$member_h = \frac{crisp - bound_1}{bound_2 - bound_1} \quad 3-2$$

where **crisp** is the crisp value found in the dataset for the **temperature** attribute, 21 in our example, **bound₁** is the value at which the boundary between the two fuzzy sets starts, 20 in our example, and **bound₂** is the value at which the boundary between the two fuzzy sets ends, 25 in our example. Substituting in the equation, we get the value of 0.2 for **member_h**.

As for the degree of membership of the crisp **temperature** value to the **mild** fuzzy set, **member_m**, the following equation can simply be used

$$member_m = 1 - member_h \quad 3-3$$

so, for the crisp value of 21, the degree of membership to the **mild** fuzzy set equals **0.8**.

The system is flexible to allow any number of fuzzy sets defined by the user for every continuous attribute and boundary points between them. As another example, we have another continuous attribute in our sample **weather** dataset, **humidity**. This attribute might be fuzzified using the fuzzy sets **high** and **normal**. The following figure shows a possible fuzzy membership function for this case

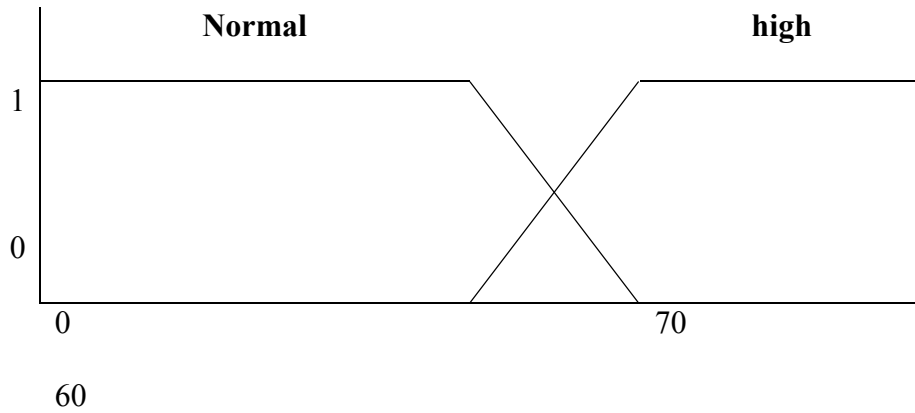


Figure 3.4 – A trapezoidal fuzzy mapping function for the humidity attribute

Calculating the degrees of membership of crisp values for the **humidity** attribute in table 3.1 to either or both fuzzy sets in figure 3.3 can take place in the same manner discussed earlier for the **temperature** attribute. For example, the first four values for **humidity** attribute in table 3.1 have membership degree of 1 to the **high** fuzzy set, and 0 to the **normal** fuzzy set since they are all above 70. The fifth row in table 3.1 carries a value of 55 for **humidity** which have membership degree of 1 to the **normal** fuzzy set, and 0 to the **high** fuzzy set since it lies below 60 where the boundary between the two fuzzy sets starts. Finally, the value of the **humidity** in the sixth row in table 3.1 is 65 which lies exactly in the middle of the boundary between the two fuzzy sets, this means equal membership degrees to both sets of **0.5**.

3-4-1 Predictive Accuracy Measure

Since the classification rules evolved by our algorithm are fuzzy rules due to fuzzification of the continuous attributes, the predictive accuracy measure used here is an extended version of

the standard accuracy rate measure extended to cope with the fuzzy nature of the evolved rules, it makes use of the standard **fuzzy AND** operator, where the degree of membership of an example dataset record i to a candidate rule antecedent is given using the minimum operator as follows

$$\text{membership}_i = \min_j^{att}(m_{ij}) \quad 3-4$$

where m_{ij} is the degree by which record i belongs to attribute j of rule antecedent, att is the number of predicting attributes in the candidate rule. If attribute j is nominal carrying a crisp value, the value of m_{ij} will be either 0 or 1, it will be 0 if the value of the attribute in record i is not equal to the value of the attribute in the candidate rule antecedent, and it will be 1 if the value of the attribute in the record is the same value appearing in the rule antecedent.

If the attribute j was a continuous fuzzified attribute, the rule antecedent will carry one of the fuzzy sets of this attribute whereas the record i will carry a crisp value for the attribute. Here, the value of m_{ij} is directly determined by the value of the degree of membership of the crisp value of the attribute in record i to the fuzzy set found in the rule antecedent, which varies between 0 and 1 as discussed earlier.

For instance, consider a rule antecedent for our sample **weather** dataset with the following attribute values:

"outlook=sunny & temperature=hot & humidity=high & windy=true",

and consider the row 2 in table 3.1, to calculate the degree of membership of this record to the sample rule antecedent, **membership₂**, we need to calculate the four values m_{21} , m_{22} , m_{23} , m_{24} which represent the degrees of membership of the record to the four predicting attribute values in the rule antecedent and then we take the minimum of them to be the value of **membership₂**. The nominal attribute **outlook** carries the value sunny in both the record and the rule antecedent, so m_{21} equals 1, for the same reason m_{24} corresponding to the **windy** attribute equals 1. For the continuous attribute **temperature**, the value in the record, 21, has a degree of membership of 0.2 to the **hot** fuzzy set existing in the rule antecedent, so m_{22} equals 0.2. Finally, the value of **humidity** attribute in the dataset has a degree of membership of 1 to the high fuzzy set existing in the rule antecedent. So, the minimum m_{2j} value is 0.2.

If a dataset record has a membership greater than 0 to a rule antecedent, and in the same time the rule consequent predicts the same value existing in the record for the goal attribute, then we can consider that this candidate rule makes a correct prediction for this record with a value equals to the record membership to the rule. We define the term **CorrPred** to measure the value of correctness of prediction of a candidate rule for all dataset records using the following equation

$$\text{CorrPred} = \sum_{i=1}^{\text{row}} \min \left[c_i, \min_j^{\text{att}} (m_{ij}) \right] \quad 3-5$$

where c_i represents the degree by which record i belongs to the rule consequent.

Now, the predictive accuracy of a candidate rule can be calculated using the following equation

$$\text{Accuracy} = \frac{\text{CorrPred} - 0.5}{\sum_{i=1}^{\text{row}} \text{membership}_i} \quad 3-6$$

The value 0.5 subtracted from **CorrPred** in the numerator of equation 3-6 can be viewed as a correction factor inserted to penalize rules that are too specific, i.e. satisfied by few training records in the dataset. For instance, suppose that we have a candidate rule satisfied by a single record in the dataset, so that CorrPred equals 1 and the denominator of equation 3-6 is also 1. Without subtracting the 0.5 from the numerator, the **accuracy** of the candidate rule will seem to be 100%, which is an over-optimistic estimate. However, subtracting the 0.5 from the numerator will return an **accuracy** estimate of 50% which might be more realistic. For large values of CorrPred and the denominator of equation 3-6, the subtraction of the 0.5 value will not have a significant influence in the value returned for the **accuracy**, this subtraction only penalizes rules covering just few training records.

3-4-2 Interestingness Measure

We use an **objective** interestingness measure in our proposed algorithm as part of the fitness evaluation of the evolved candidate classification rules. As discussed in the previous chapter, there are two categories of interestingness measures, **objective** and **subjective** measures.

Objective measures tend to be more general and autonomous than subjective measures are. We would like to emphasize here that an operating data mining algorithm should use both kinds of measures whenever possible. An objective measure might be the basis for the interestingness evaluation of candidate rule, where a domain-dependent subjective measure might make use of the user's domain knowledge to strengthen the objective measure.

A candidate classification rule has two parts, the **antecedent** and **consequent**. Our interestingness measure is derived from both rule parts; **AntInt** (measure of the interestingness of the rule antecedent) and **ConsInt** (measure of the interestingness of the rule consequent). So overall interestingness can be measured as

$$Interestin\ gness = \frac{AntInt + ConsInt}{2} \quad 3-7$$

The methods used to calculate the rule antecedent interestingness **AntInt** and the rule consequent interestingness **ConsInt** are modified fuzzy versions of the methods used by Noda et al [20].

As for the rule antecedent, the more unexpected attributes appear in the rule antecedent, the more interesting the rule is. So **AntInt** is high for rules containing attributes with low information gain. These attributes are probably considered by the user to be irrelevant, and they are kind of irrelevant for classification when considered individually, one at a time. However,

attribute interaction can render an individually irrelevant attribute into a relevant one, and this phenomenon is intuitively associated with rule interestingness [20].

$$AntInt = 1 - \sum_{i=1}^{att} \frac{InfoGain(A_i)}{att * \log_2(dom(G))} \quad 3-8$$

Equation 3-8 shows how **AntInt** is calculated for a candidate rule antecedent based on the idea stated in the previous paragraph, where **att** is the number of attributes in the rule antecedent, **InfoGain(A_i)** is the information gain of attribute i in the rule antecedent and **dom(G)** is the domain cardinality, number of values, of the goal attribute. The term subtracted from the value of 1 in the equation represents a normalized average of the information gains of attributes in the rule antecedent, as this value increases, meaning presence of attributes with high **InfoGain**, the value of **AntInt** decreases, and vice versa. The log term in the denominator is to insure a normalized value of the average information gain. The information gain of an attribute can be calculated from data using probabilistic terms as in the following equations

$$InfoGain(A_i) = Info(G) - Info(G \mid A_i) \quad 3-9$$

To understand the implication of the information gain of a predicting attribute, consider how it is calculated in equation 3-9. **Info(G)** is the information content of the goal attribute measured using the concepts of information theory as will be shown in subsequent equation.

Info(G | A) is the information of the goal attribute given that the value of the predicting attribute A is known.

Now, if attribute A is considered to be an irrelevant poor predictor of G, then the value of **Info(G | A)** will not go far from the value of **Info(G)** since the given value of A is not relevant. Subtracting the two terms will give a small value for **InfoGain(A)**. On the contrary, if attribute A is known and expected to be a good predictor of G, then if the value of A is given, the information of the goal will decrease so the difference between **Info(G)** and **Info(G | A)** will be large so **InfoGain(A)** will be a large value.

The calculation of **Info(G)** and **Info(G | A)** is driven from the data probabilistically as in the two following equations

$$Info(G) = - \sum_{i=1}^{dom(G)} (\Pr(g_i) \log_2 (\Pr(g_i))) \quad 3-10$$

where g_i is the i^{th} value of the goal attribute.

$$Info(G | A_i) = - \sum_{j=1}^{dom(A_i)} \left(\Pr(V_{ij}) * \left(\sum_{k=1}^{dom(G)} (\Pr(g_{V_{ij}}^k) \log_2 (\Pr(g_{V_{ij}}^k))) \right) \right) \quad 3-11$$

About measuring the interestingness of the rule consequent, we consider that the rarer the class value, the more interesting a rule predicting it is. This means that **ConsInt** will be high for less frequent goal values. For example, considering the sample **weather** dataset of table 3.1, the class attribute **play** has two possible values; **yes** and **no**. Fewer instances of the pair **play=no**

exist in the table, 5 out of the 14 records, so rules predicting **play=no** will have higher **ConsInt** value than rules predicting the class value **play=yes**. **ConsInt** is inversely proportional with the goal attribute value present in the rule consequent; it is calculated using the following equation

$$ConsInt = \sqrt{1 - \Pr(g_i)} \quad 3-12$$

where g_i is the goal value in the rule consequent. Calculating the value of **ConsInt** for rules with goal attribute **play** in our sample **weather** dataset; for candidate rules with consequent **play=no**, **ConsInt** equals **0.802**, while rules with consequent **play=yes** have a **ConsInt** value of **0.598**.

3-5. Genetic operators

For the proposed genetic algorithm, the following genetic operators are used:

- The selection scheme used is the tournament selection.
- We use single point crossover with crossover probability **0.7**.
- We mutate individual's genes with a probability of **0.05**, the mutation operator is also used to remove or insert attributes to a candidate rule as will be soon described.
- An elitism factor of one is used to keep the best candidate rule reached in every generation.

- The main experiment of the algorithm set the population size to **300** and the number of generations to **600**.
- An important aspect of the proposed algorithm is the way it manages the size of the candidate rules evolved. Instead of using special operators for insertion and removal of attributes from the rules, we let the initialization and mutation operators do the job using a simple scheme that showed similar results to the case when special operators were used for attribute insertion and removal. During initialization of the gene of an attribute, a probability of **0.5** is associated with the attribute removal from the rule by putting the flag value of **-1** in the gene, and a probability of **0.5** is associated with the natural initialization of the gene with an index of a value from the domain of the attribute. The mutation operator is also modified to handle removal of attributes, if a gene is to be mutated, probability of this is **0.05** as stated earlier, the new value of the mutated gene will be **-1** with a probability of **0.5**, and will be another value from the domain of the attribute with a probability of **0.5**. So, the probability that an attribute be removed by mutation is **0.5*0.05** which is **0.025**. Also mutation can insert a removed attribute by changing the **-1** flag value of its gene to a value from the domain of the inserted attribute, the probability of this event is the ordinary mutation probability which is **0.05** in our genetic parameter settings.

CHAPTER 4

Classification Experiments

4-1 Gait Data Collection

The subjects for the experiments of this thesis were twenty three healthy adults (twelve females and eleven males) recruited from the campus of the University of Texas at El Paso (UTEP) after having given their written informed consent. These subjects reported no physical or mental disorders, and no use of any medications at the time of data collection. Table 4-1 shows more details about the participating subjects.

The data collection experiments took place at the Laboratory for Human Motion Analysis and Rehabilitation located in the UTEP Electrical and Computer Engineering Department. The laboratory is equipped with:

- a- A camera system (eight high resolution cameras) to capture video data of the subjects while walking from eight different views.
- b- An instrumented treadmill with separate left and right force plates to capture the ground reaction forces of both sides of the body while walking (Fig. 4.1).
- c- A 16-channel electromyography(EMG) system with electrodes connected to selected muscles of a subject to capture his/her muscle activity while walking. The 16 electrodes were used to record the activity of 16 muscles; 8 on each side. For this study, the EMG activity of four muscles on the left leg was considered. These muscles are the soleus, tibialis anterior, gastrocnemius, and vastus lateralis.
- d- A software system (SIMI Motion Analysis) to manage capturing and synchronizing the three different types of data.

Table 4.1 – Subjects participating in gait data collection experiments for this thesis

Subject ID	Gender	Age	Height (cm)	Weight (kg)	Body Mass Index	Data Used for
M1	Male	27	171	59	20.2	Training
M2	Male	29	190	88	24.4	Training
M3	Male	37	171	87	29.8	Training
M4	Male	23	184	112	33.1	Training
M5	Male	50	176	69	22.3	Training
M6	Male	40	185	98	28.6	Training
F1	Female	22	168	58	20.5	Training
F2	Female	23	154	40	16.9	Training
F3	Female	23	184	81	23.9	Training
F4	Female	27	161	63	24.3	Training
F5	Female	25	168	50	17.7	Training
F6	Female	47	152	83	35.9	Training
F7	Female	21	168	61	21.6	Training
M7	Male	24	167	77	27.6	Testing
M8	Male	26	175	80	26.1	Testing
M9	Male	58	171	76	26	Testing
M10	Male	37	181	76	23.2	Testing
M11	Male	19	167	66	23.7	Testing
F8	Female	37	157	55	22.3	Testing
F9	Female	23	158	62	24.8	Testing
F10	Female	22	161	46	17.7	Testing
F11	Female	28	152	47	20.3	Testing
F12	Female	19	169	63	22.1	Testing

Each subject performed three walking experiments, the first was at the natural walking speed of the subject, and the second was at a 20% slower speed, while the third was at a 20% higher speed. The natural walking speed of a subject was estimated by letting the subject walk on the treadmill at different speeds for some time until he felt he reached his natural speed. In each experiment, the subject walked for three minutes and the first 120 strides were extracted to be used by the classification algorithms. Repeating the experiment at three different speeds was performed to study how gait variables change with the walking speed, and whether the way they change differs from one subject to another, so for this thesis, the walking speed was used as one of the predicting attributes.



Figure 4.1 – Specialized treadmill used for collecting ground reaction forces of both sides of the body while walking.

The following attributes were extracted for every stride in each of the three experiments done for every subject and were used as predicting attributes for the classification experiments:

- 1- Speed of walking: Three nominal values were used; 1 for the slower speed, 2 for the natural speed, and 3 for the faster speed.
- 2- Stride time: Time for one stride which is the time between two consecutive heel strikes of one leg [23, 24].
- 3- Stride length: Distance in sagittal plane between the heels from one heel strike to the next heel strike of the same leg [23, 24].
- 4- Stride width: Distance in transverse plane between the heels from one heel strike to the next heel strike of the same leg [23, 24].
- 5- Stance/Swing ratio: Ratio between the stance time and the swing time within one gait cycle.
- 6- Double stance phase ratio: Ratio between the time when both legs are in stance phase and the stride time.
- 7- Ground Reaction Force on left plate of the treadmill: The force on the left plate of the treadmill for three strides of an experiment is shown in Fig. 4.2. It can be seen that during the stance phase, three peaks can be found. Only one feature was extracted and used as a predicting attribute for the classification experiments, which is the average value of these peaks. Before extracting this feature for every gait cycle, a second order low pass filter was applied to the force signal to smooth it.

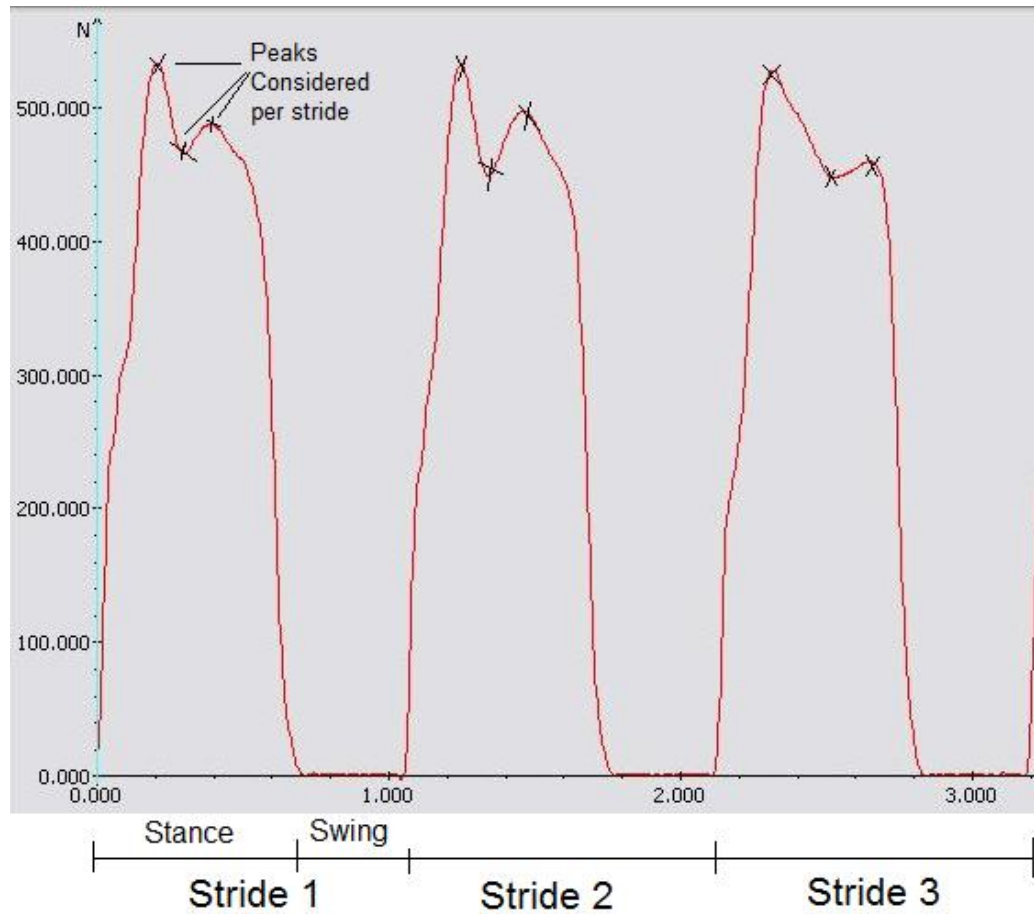


Figure 4.2 - The z-component of the force on the left plate for 3 strides, showing the feature extracted per stride which is the average of the three peaks

- 8- Left soleus muscle activity: For each stride, the EMG signal from the left soleus muscle was captured. By inspecting the average muscle activity for all collected strides of all subjects, it was noticed that the highest peak voltage occurred at a point after approximately 45% of the gait cycle. For each stride, the extracted feature that was considered as a predicting attribute for the classification experiments was the percent shift of the highest peak point from the average (45%) and the direction of that shift (right or left). An example is shown in Fig. 4.3 for a typical stride from an experiment of one of the participating subjects.

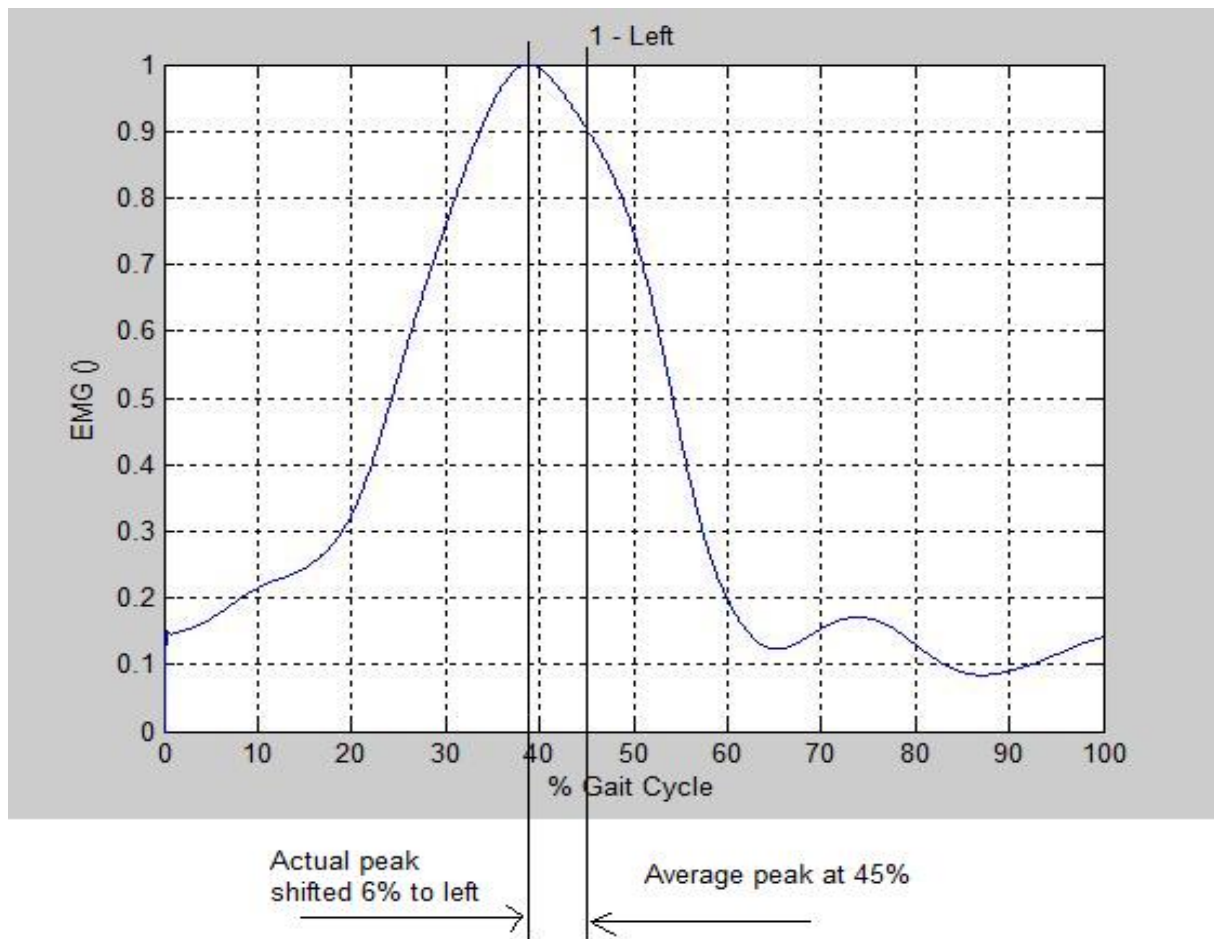


Figure 4.3 - Normalized EMG activity of soleus muscle during one gait cycle of an experiment, where the peak is shifted to the left compared to its average point.

- 9- Left tibialis anterior muscle activity: Similar to the previous muscle, the maximum peak was found to lie on the average at 8% from the beginning of the gait cycle, the extracted feature was also the percent shift from this average peak and the direction of that shift. An example is shown in Fig. 4.4 for a typical stride from an experiment of one of the participating subjects.

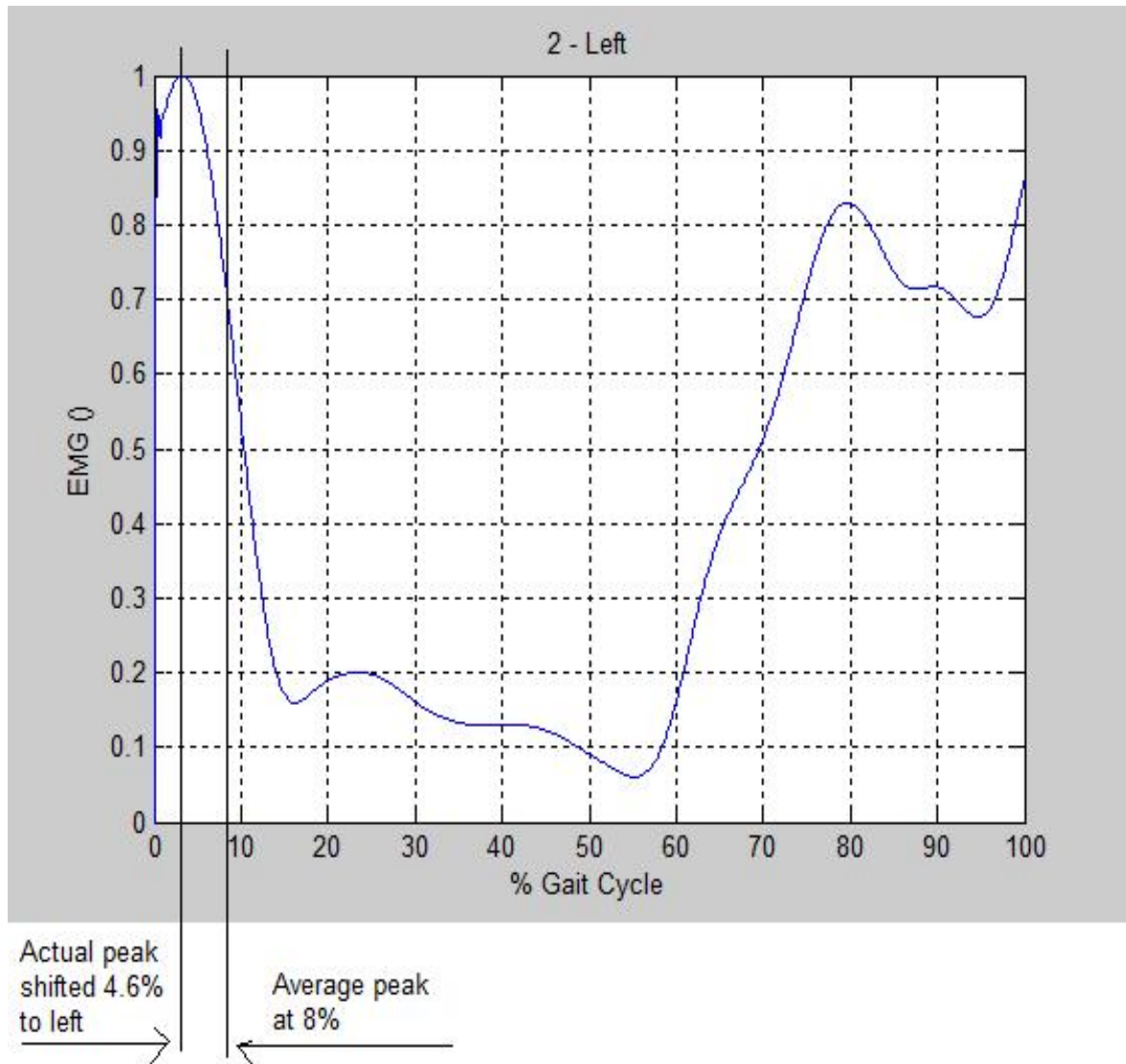


Figure 4.4 - Normalized EMG activity of tibialis anterior muscle during one gait cycle of an experiment, where the peak is shifted to the left compared to its average point.

10- Left gastrocnemius muscle activity: Following the same procedure of the previous muscles, the extracted feature for this muscle was the percent shift and its direction from the average peak at 40% from the beginning of the gait cycle. An example is shown in Fig. 4.5 for a typical stride from an experiment of one of the participating subjects.

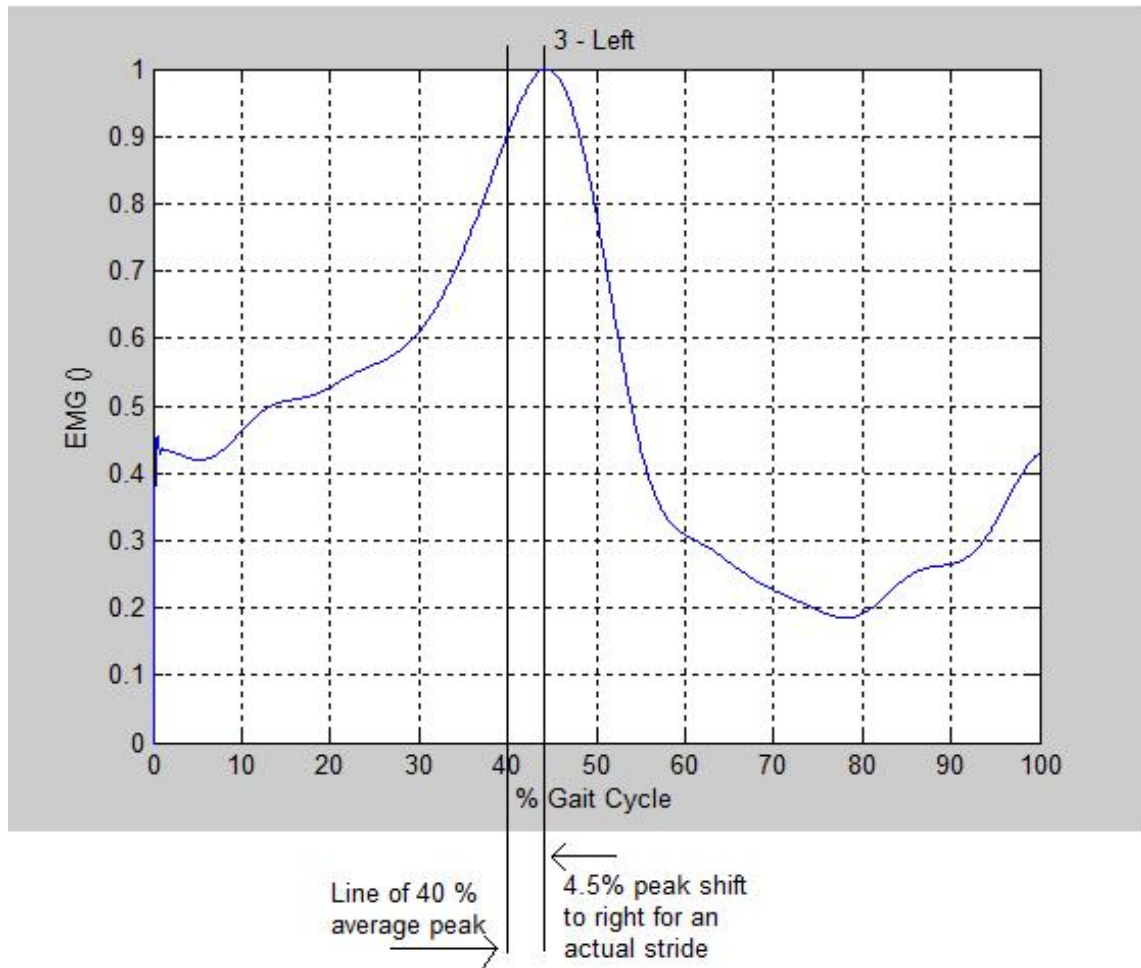


Figure 4.5 - Normalized EMG activity of gastrocnemius muscle during one gait cycle of an experiment, where the peak is shifted to the right compared to its average point.

11-Left vastus lateralis muscle activity: Following the same procedure of the previous muscles, the extracted feature for this muscle was the percent shift and its direction from the average peak at 10% from the beginning of the gait cycle. An example is shown in Fig. 4.6 for a typical stride from an experiment of one of the participating subjects.

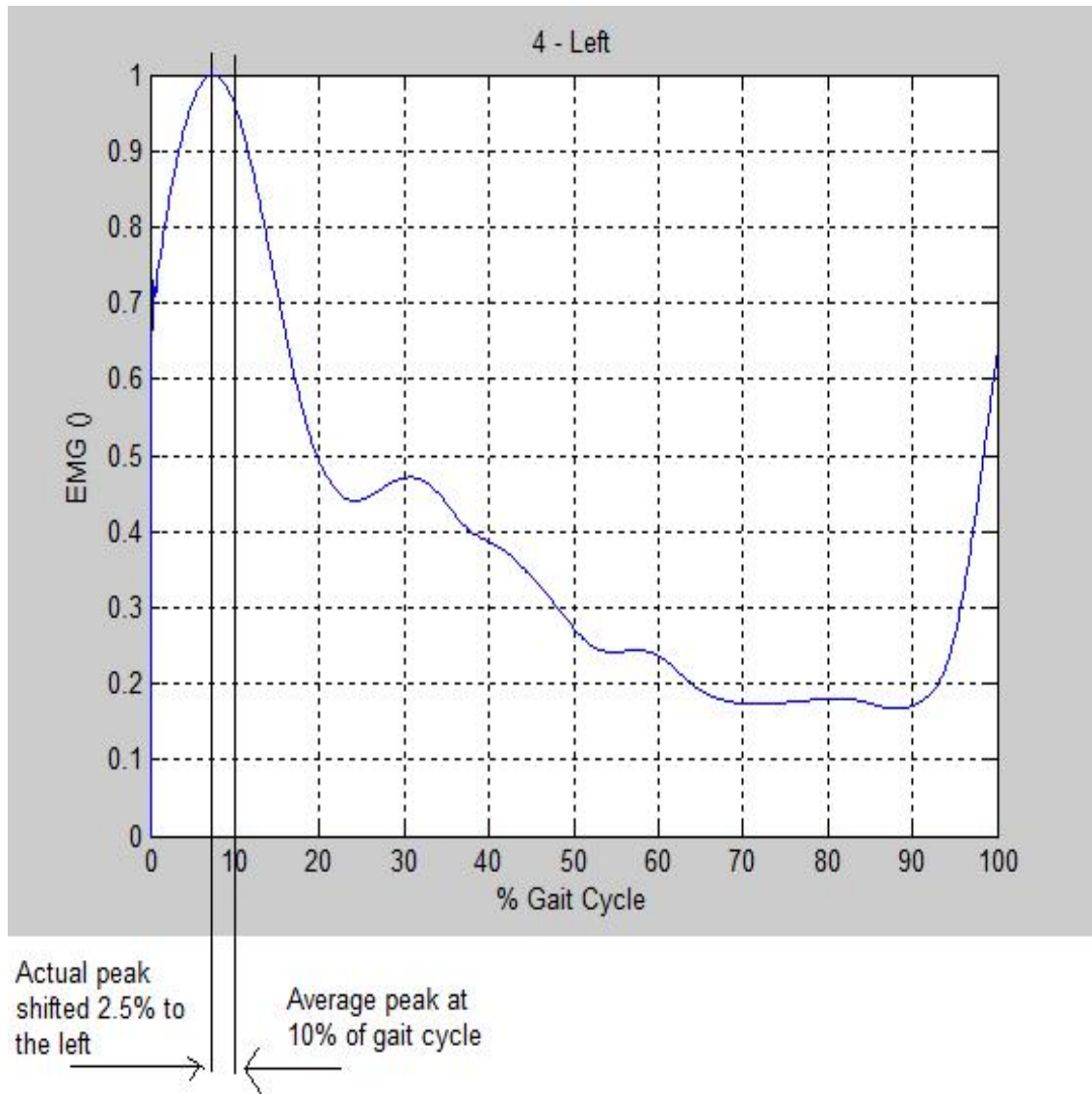


Figure 4.6 - Normalized EMG activity of vastus lateralis muscle during one gait cycle of an experiment, where the peak is shifted to the left compared to its average point.

12- Body mass index (BMI) of the subject

13- Height of the subject.

14- Age of the subject.

15- Gender of the subject: Two nominal values were used; 1 for male subjects, and 2 for female subjects.

4-2 Fuzzification of Continuous Attributes

Except for the first and last predicting attributes mentioned at the end of the previous section, all other considered attributes are continuous. Before feeding any row of the dataset to any tested classification algorithm, the crisp values of these attributes need to be fuzzified first. For each attribute, three fuzzy sets were defined where the definitions were based on the average values of this attribute for all strides of all participating subjects. Fig. 4.7 through Fig. 4.19 show the fuzzy membership functions defined for the 13 continuous attributes.

The defined membership functions were either trapezoidal or triangular (which can be viewed as a special form of the trapezoidal function).

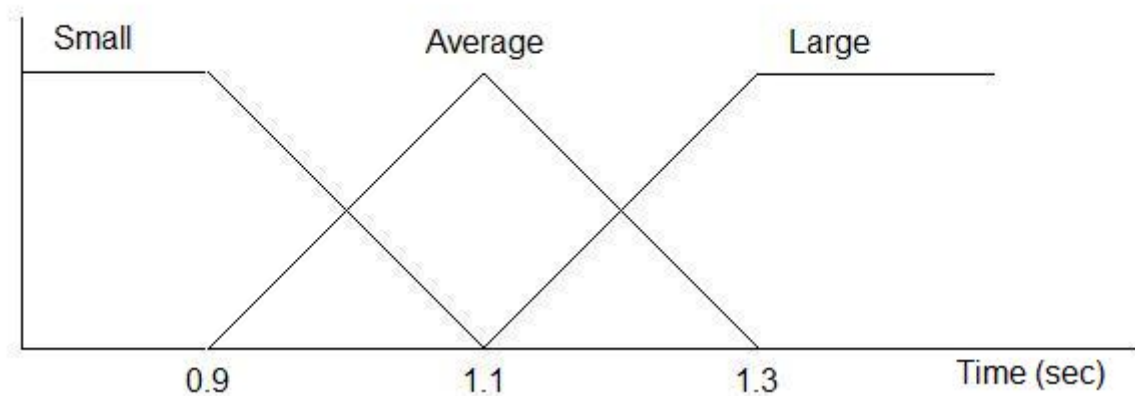


Figure 4.7 – Stride time fuzzy membership function

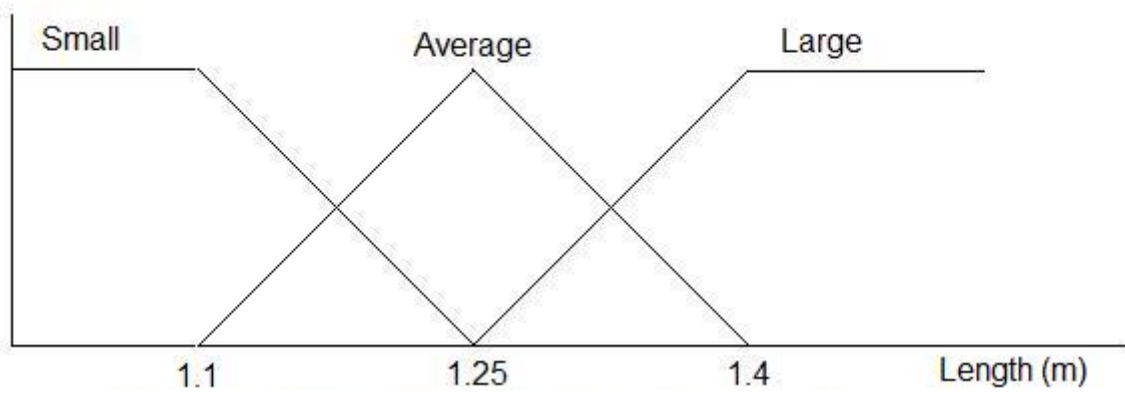


Figure 4.8 – Stride length fuzzy membership function

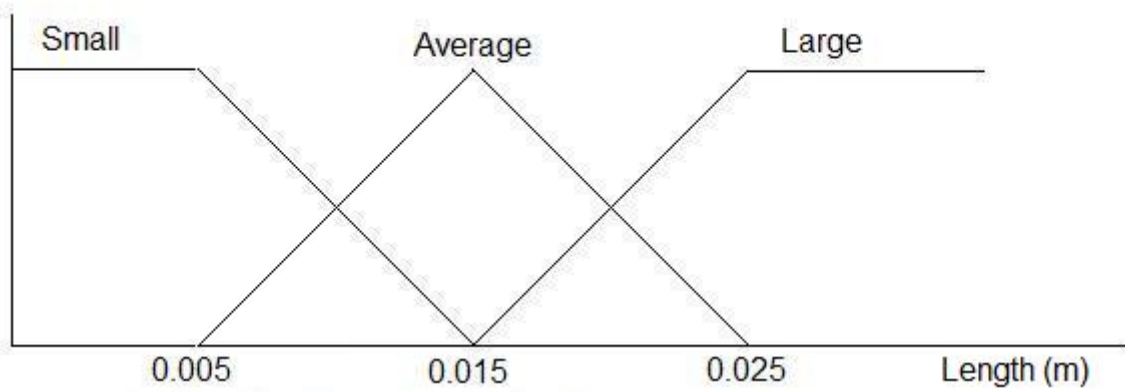


Figure 4.9 – Stride width fuzzy membership function

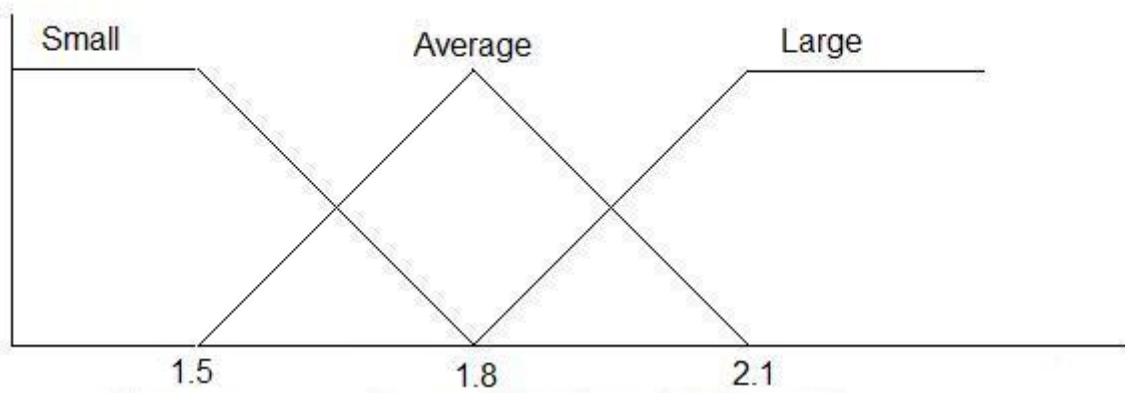


Figure 4.10 – Stance/Swing fuzzy membership function

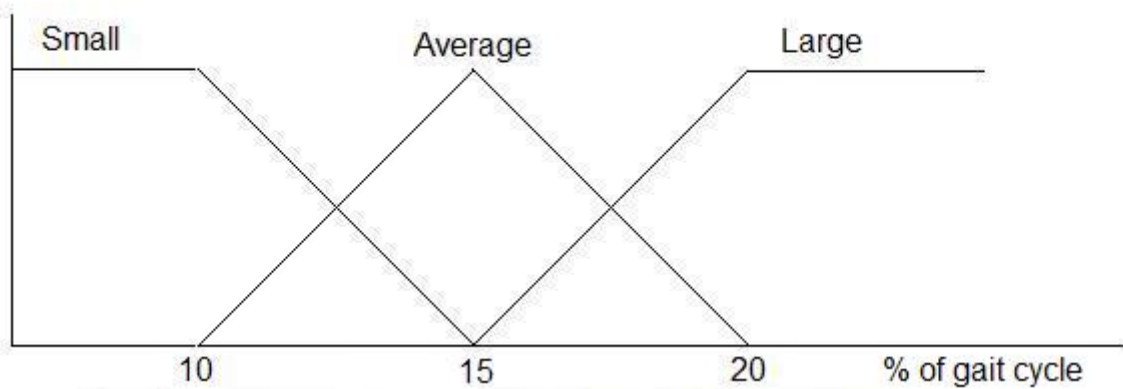


Figure 4.11 – Double stance phase ratio fuzzy membership function

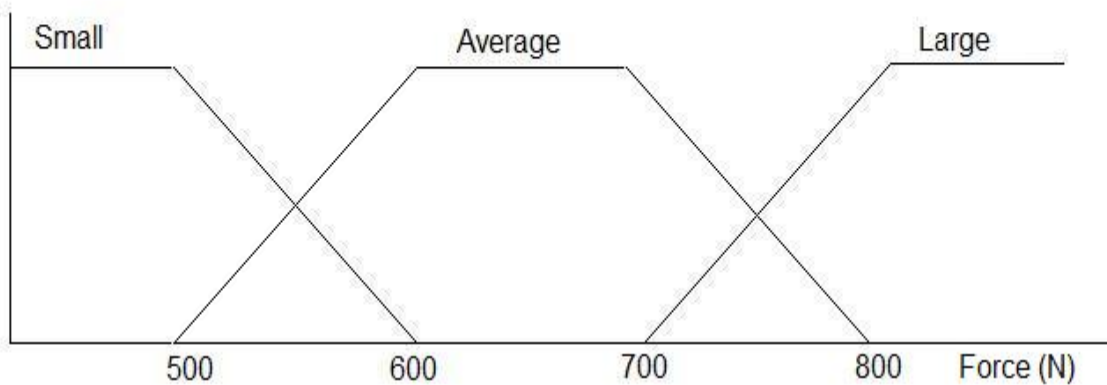


Figure 4.12 – Left force plate feature fuzzy membership function

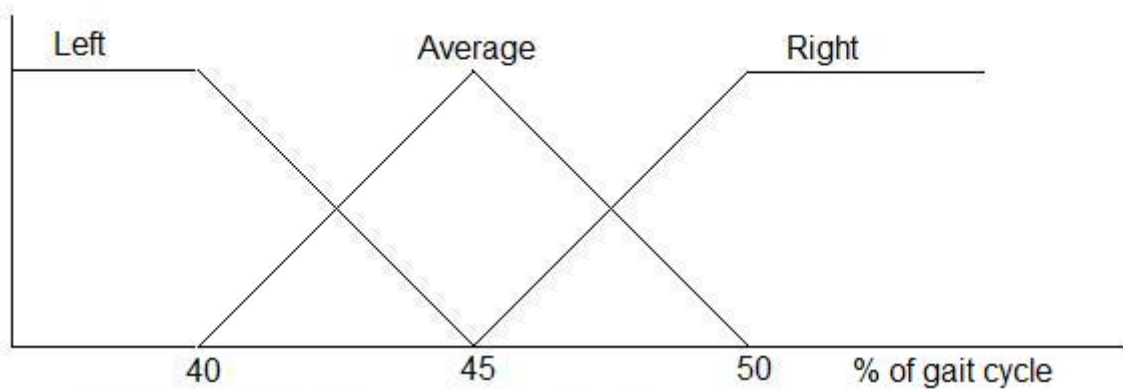


Figure 4.13 – Soleus peak shift fuzzy membership function

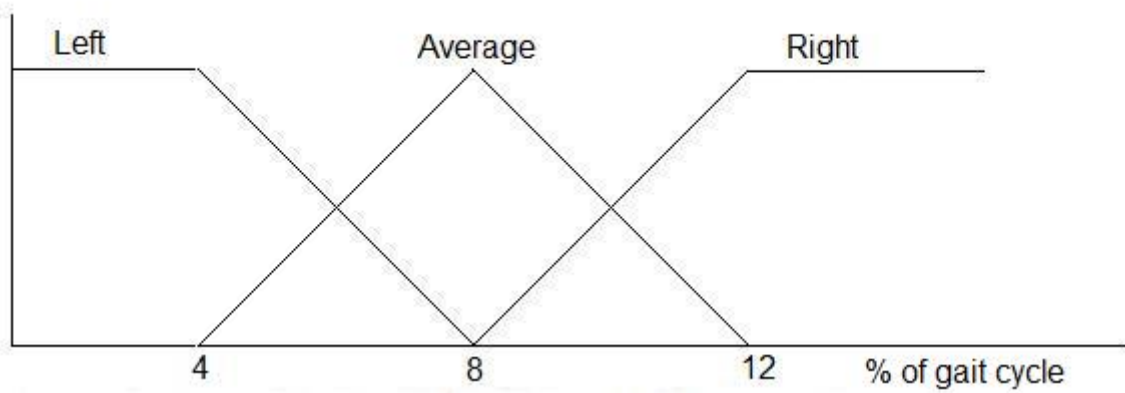


Figure 4.14 – Tibialis Anterior peak shift fuzzy membership function

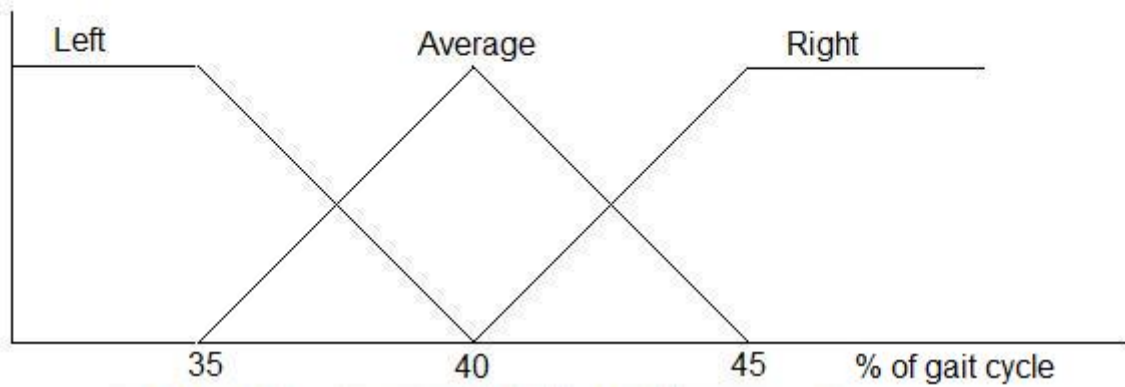


Figure 4.15 – Gastrocnemius peak shift fuzzy membership function

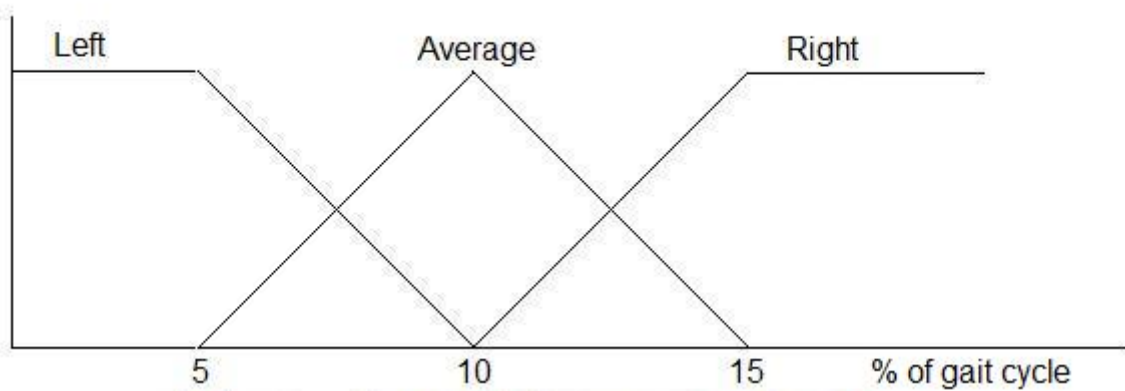


Figure 4.16 – Vastus Lateralis peak shift fuzzy membership function

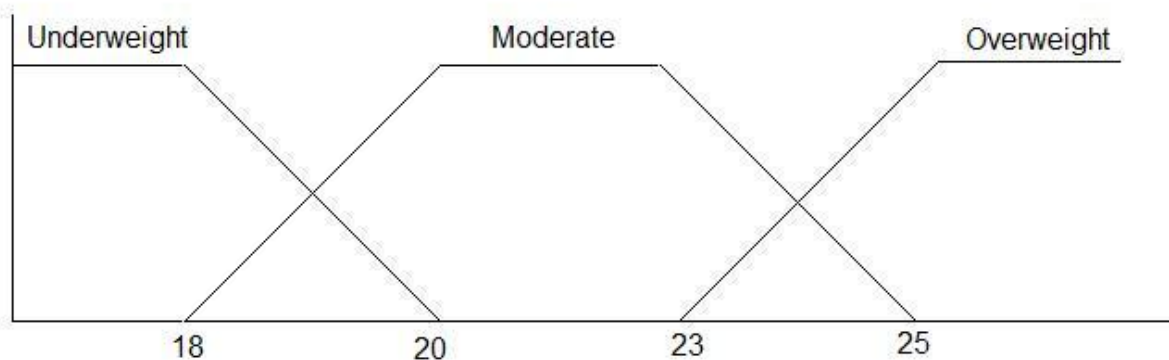


Figure 4.17 – Body mass index fuzzy membership function

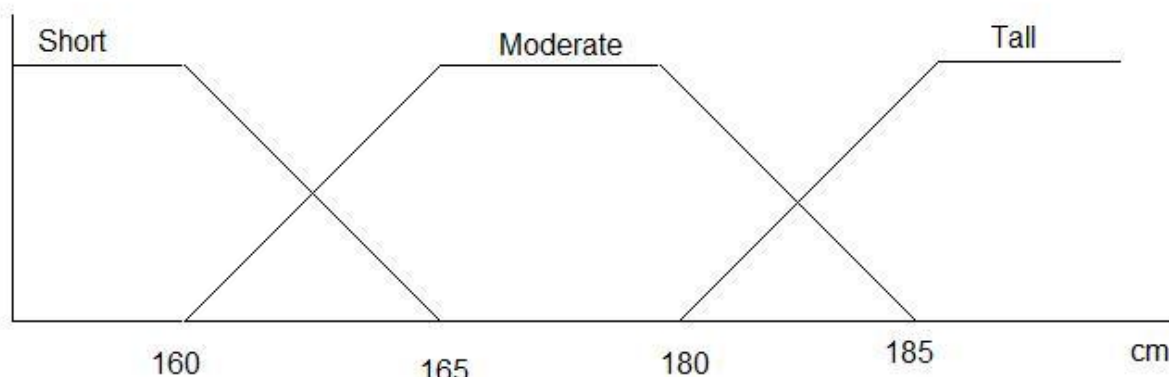


Figure 4.18 – Height fuzzy membership function

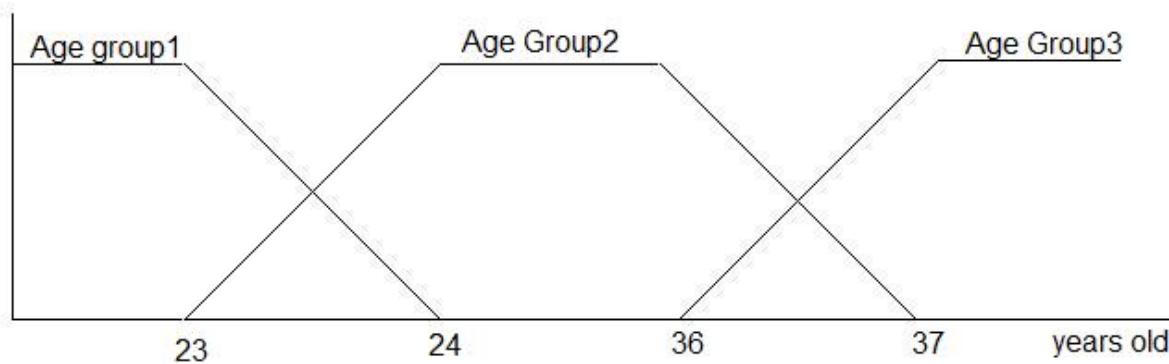


Figure 4.19 – Age fuzzy membership function

It is worth noting that these defined membership functions can be enhanced in a number of ways. For simplicity and consistency, only three fuzzy sets were defined for each continuous attribute. Careful study of the distribution of values of each attribute especially with a bigger dataset might lead to definition of more fuzzy sets per attribute. Defining the membership functions in terms of the standard and simple trapezoidal and triangular membership function was first made for simplicity, but it was also aimed to develop this work to study how well a fuzzy genetic algorithm will fit to the different classification tasks based on human gait data. For the same reason, the three fuzzy sets for each attribute were defined based on the distribution of the data values for this attribute for all the data collected from the experiments of the 23 participating subjects. The membership function of the second fuzzy set of an attribute was almost centered around the average value of this attribute in the dataset. The boundaries for the fuzzy sets were defined in such a way that the values of the attribute in the dataset be distributed as evenly as possible among the three fuzzy sets of the attribute.

4-3 Gait Identification Experiments

The objective of the experiments discussed in this section is to test if the proposed fuzzy genetic algorithm can discover classification rules that can really identify a subject given the values of predicting attributes of one stride or a set of strides (experiment).

The dataset used for this task was taken from thirteen subjects (7 females and 6 males). The number of walking experiments considered here are 37 experiments (12 fast walking, 12 slow walking, and 13 natural speed walking), the reason for the less fast and slow experiments than the natural ones was due to the availability of only the natural speed walking data for one of the 13 subjects of this task. The predicting attributes were all the attributes mentioned in section 4-1 except for the age and gender that were not used for this experiment.

The age distribution of the subjects in these experiments was as follows; five subjects belonged to the younger age group (18-24 years old), four subjects were in each of the other two age groups. The BMI distribution of the subjects in these experiments was as follows; seven subjects were of moderate BMI (18-24), four subjects were overweight, and two subjects were underweight.

The following procedure was followed:

- 1- For the 120 strides captured from each walking experiment, 80 strides were used for training of the tested genetic algorithms, while the remaining 40 strides were used to test the ability of the discovered rule sets to identify the subject. Thus, the size of the training

dataset was 2960 rows (strides) and 14 attributes, while the size of the test dataset was 1480 rows and 14 attributes.

- 2- Three variants of the proposed GA were experimented with the training dataset to study the effect of introducing an interestingness measure in the fitness function and to study the effect of varying the weights of the interestingness and predictive accuracy measures in the fitness function of the genetic algorithm. The difference between the three algorithms lied then in the weights of the fitness function.

$$fitness = \frac{w1 * Interestingness + w2 * Accuracy}{w1 + w2}$$

For the first algorithm, $w1=0$ and $w1=1$, meaning that only the predictive accuracy of the tested candidate rule is considered. For the second algorithm, $w1=1$ and $w2=2$, meaning that the interestingness measure is considered but the weights are fixed, the same weight values were used by Noda et al [20]. The third algorithm is using adapting weights for both elements of the fitness function. At the beginning of the GA run, $w1=1$ and $w2=0$ as in Fig. 4.20, to encourage the GA to search for interesting rules. As the GA proceeds, $w1$ decreases gradually while $w2$ increases to support the accurate rules among the explored interesting discovered rules. At the end of the GA run, $w1=0$ and $w2=1$.

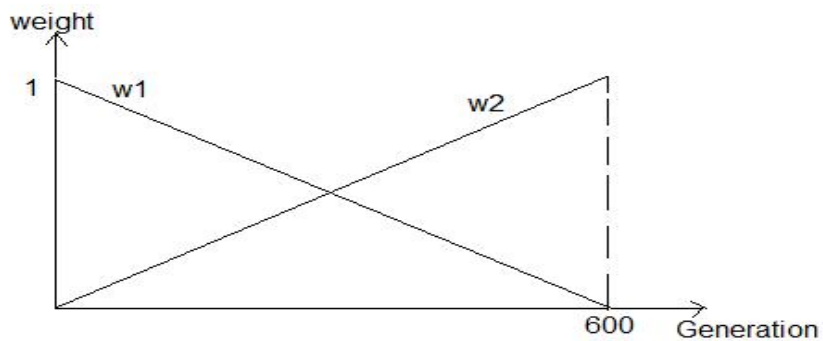


Figure 4.20 – Adapting weights for the genetic algorithm

- 3- A separate GA run was used to discover the fuzzy classification rule set identifying one subject. This means for every variant of the proposed GA, thirteen runs were executed to get the thirteen rule sets used to identify the subject of every row in the test dataset.
- 4- For each rule set resulting from a GA run, only the best 40 distinct rules in terms of the value of the fitness function are considered, in all cases, those 40 rules were always having fitness values above 0.9.
- 5- To identify the subjects based on an input stride, the identification algorithm was run by inputting a row (representing a stride) from the test dataset to all the rule sets identifying different subjects. The continuous values of the predicting attributes in the input row were fuzzified then each rule set evaluated the input row by counting the number of fuzzy rules in the set that were fully matching the input fuzzified row. The rule set of the subject with the highest count of the matched rules will be the prediction of the subject ID for the input row (stride). The process is shown in Fig. 4.21.
- 6- To identify the subjects based on an experiment (set of 40 strides belonging to the same subject), the previous step was repeated for the 40 strides of an experiment, and the prediction of the subject ID for the experiment will be the predicted subject for the majority of the individual strides. The process is shown in Fig. 4.22.
- 7- Each of the two previous steps was repeated four times, considering different number of rules per rule set to study the effect of the number of rules considered on the hit ratio for

each experimented variant of the GA. The studied sizes of the rule sets were 10, 20, 30, and 40 rules per rule set.

- 8- The resultant rule sets from the second and third GA variants were merged to study the effect of combining rules from different methods on the hit ratio of the identification algorithm, two sizes of the combined rule sets were studied; 10 and 20 rules per rule set.

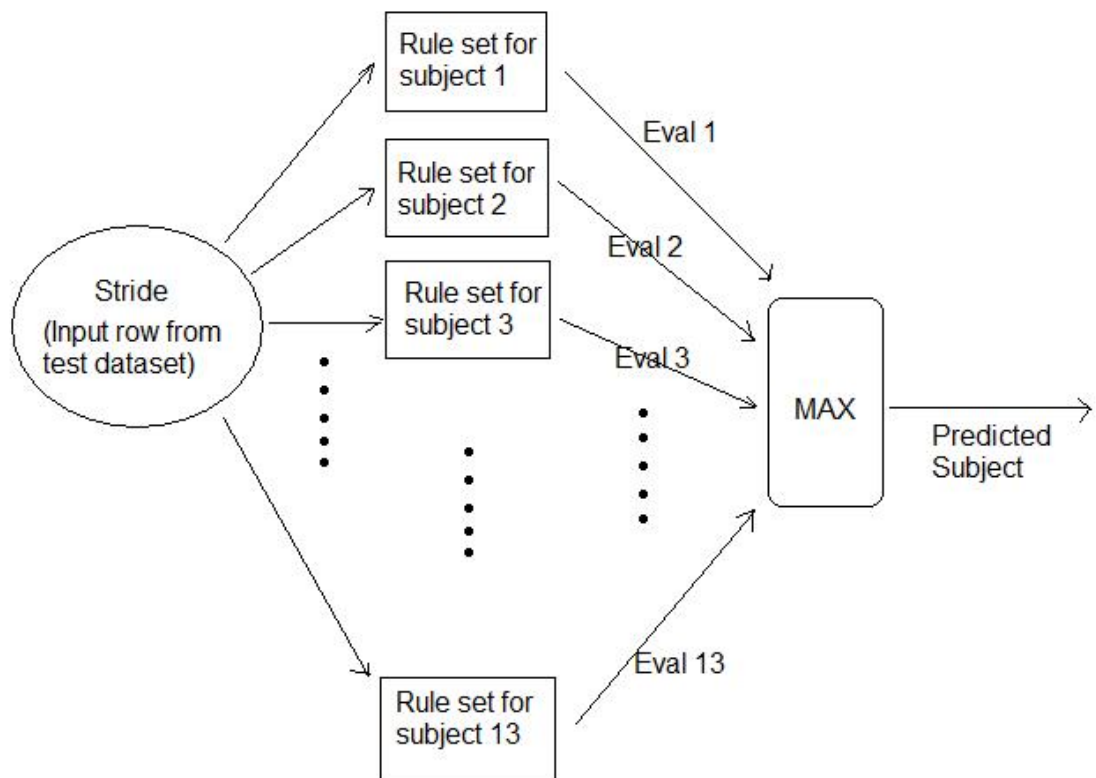


Figure 4.21 – Identifying the subject of a stride using rule sets of all subjects

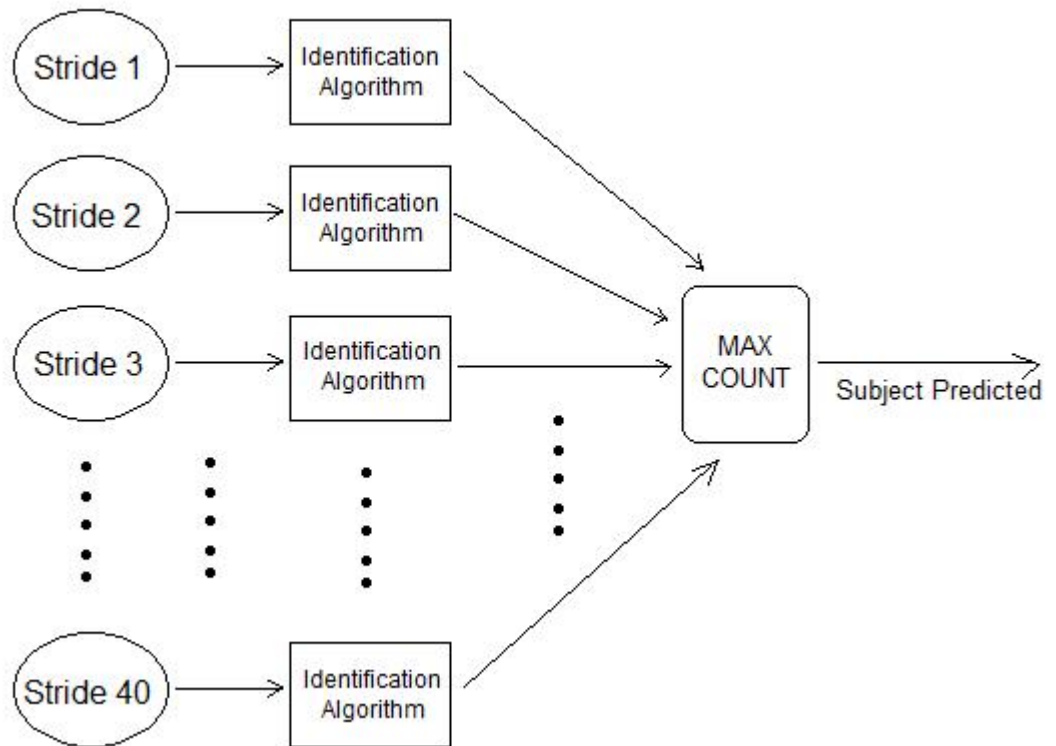


Figure 4.22 – Identifying the subject based on an experiment (set of strides)

4-4 Gait Similarity Experiments

The objective of the experiments discussed in this section is to test if the proposed fuzzy genetic algorithm can discover classification rules that can be used to map the data of a new subject to find the closest subject in the training dataset to the new subject.

The same dataset used for training in section 4-3 was used for training here. The data for the ten subjects in Table 4.1 that were not used in the gait identification experiments (section 4-3) were used for testing the capability of the fuzzy GA to map the new subjects to similar subjects in the training dataset. The test dataset was composed of 24 experiments (10 natural walking experiments, 7 slow walking experiments, and 7 fast walking experiments).

A simple measure for similarity between two subjects was defined to equal the number of attributes that have the same fuzzy value for both subjects divided by the total number of attributes considered. All attributes in section 4-1 were considered for this measure except for the speed of experiment. Table 4.2 shows the fuzzy values for these attributes for the 23 participating subjects. Using the values in Table 4.2 to calculate the similarity measures for every pair of subjects yielded the similarity matrix shown in Table 4.3 which was used to evaluate the results for running the similarity algorithm.

The same three GA variants discussed in section 4.3 were run where the input was the 120 strides from each of the 24 experiments in the test dataset. The four sizes of rule sets were experimented for each GA variant, and the same manner of combining the rule sets of fixed and adapting weights was followed.

Table 4.2 – Fuzzy values of attributes of all subjects

Subject	Gender	Age	Height	BMI	Str. Len.	Str. Wid.	Str. Time	St / Sw	Double St	LFP	Soleus	Tibialis	Gastro.	Vastus
M1	1	2	2	2	2	1	2	2	2	2	3	2	3	2
M2	1	2	3	3	3	2	3	3	3	3	2	1	3	2
M3	1	3	2	3	2	2	3	2	2	2	2	1	1	1
M4	1	1	3	3	3	3	3	3	2	3	3	1	1	1
M5	1	3	2	2	3	2	2	2	2	2	2	2	1	1
M6	1	3	3	3	3	2	3	2	2	3	2	1	3	1
F1	2	1	2	2	2	2	2	3	3	2	3	2	2	2
F2	2	1	1	1	2	2	2	2	1	1	3	2	3	1
F3	2	1	3	2	3	3	2	2	2	3	2	1	3	2
F4	2	2	1	3	1	2	2	2	2	2	1	2	2	2
F5	2	2	2	1	1	2	2	2	2	1	2	1	3	2
F6	2	3	1	3	1	2	1	3	2	3	1	1	2	2
F7	2	1	2	2	2	2	3	3	2	2	1	1	2	2
M7	1	2	2	3	2	1	2	3	3	2	3	1	2	2
M8	1	2	2	3	3	3	3	3	2	3	3	1	2	2
M9	1	3	2	3	3	2	2	2	2	2	1	2	3	2
M10	1	3	3	2	3	1	3	2	2	2	2	2	3	2
M11	1	1	2	2	2	1	2	2	2	2	3	1	3	2
F8	2	3	1	2	1	1	1	2	2	1	1	2	3	2
F9	2	1	1	3	2	1	2	2	2	2	3	2	2	1
F10	2	1	1	1	1	2	2	3	2	1	1	1	3	1
F11	2	2	1	2	2	2	2	1	2	1	3	1	2	3
F12	2	1	2	2	2	2	2	3	2	2	3	2	3	3

Table 4.3 – Similarity matrix between training (rows) and testing (columns) subjects

	M7	M8	M9	M10	M11	F8	F9	F10	F11	F12
M1	64.3%	35.7%	64.3%	64.3%	85.7%	50%	57.1%	21.4%	42.9%	64.3%
M2	50%	64.3%	42.9%	50%	28.6%	14.3%	7.1%	28.6%	28.6%	21.4%
M3	42.9%	42.9%	57.1%	50%	50%	21.4%	42.9%	28.6%	28.6%	35.7%
M4	35.7%	71.4%	28.6%	35.7%	35.7%	7.1%	35.7%	35.7%	21.4%	21.4%
M5	28.6%	28.6%	71.4%	64.3%	50%	35.7%	42.9%	28.6%	28.6%	50%
M6	21.4%	50%	57.1%	64.3%	35.7%	28.6%	28.6%	28.6%	21.4%	21.4%
F1	64.3%	35.7%	42.9%	28.6%	57.1%	28.6%	57.1%	35.7%	50%	78.6%
F2	21.4%	7.1%	35.7%	21.4%	42.9%	42.9%	64.3%	64.3%	50%	57.1%
F3	21.4%	42.9%	42.9%	57.1%	57.1%	42.9%	35.7%	42.9%	35.7%	42.9%
F4	42.9%	35.7%	64.3%	35.7%	35.7%	57.1%	64.3%	50%	50%	42.9%
F5	35.7%	35.7%	50%	35.7%	50%	50%	28.6%	64.3%	50%	42.9%
F6	35.7%	50%	42.9%	21.4%	21.4%	57.1%	35.7%	57.1%	42.9%	28.6%
F7	50%	50%	42.9%	35.7%	57.1%	35.7%	42.9%	50%	50%	64.3%

4-5 Gender Classification Experiments

The objective of the experiments discussed in this section is to test if the proposed fuzzy genetic algorithm can discover classification rules that can classify the gender of a subject given the values of predicting attributes of one stride or a set of strides (experiment). The predicting attributes were all the attributes mentioned in section 4-1 except for the gender which is the goal attribute. The dataset used for this task was taken from all the twenty-three participating subjects. As was shown in Table 4.1, the data for thirteen subjects formed the training dataset while the data from the remaining ten subjects formed the test dataset. Table 4.4 shows the properties of the subjects in both datasets.

Table 4.4 – Properties of subjects in Training and Test datasets

		Training	Testing
Gender	Males	6	5
	Females	7	5
Age Group	1 (<24 y)	5	4
	2 (24-35 y)	4	3
	3 (>35 y)	4	3
BMI	Underweight	2	1
	Natural	7	6
	Overweight	4	3
Speed	Slow	12	7
	Natural	13	10
	Fast	12	7

A procedure very similar to the one discussed in section 4-3 was followed. The three variants of the proposed GA were experimented to study the effect of the interestingness measure. All the 120 strides of the experiments of subjects in the training set were used as input to the GAs. The size of the training dataset was 4440 rows (strides) and 15 columns (attributes). The size of the test dataset was 2880 rows and 15 attributes.

A separate GA run was used to discover the fuzzy classification rule sets for the two genders. This means for every variant of the proposed GA, two runs were executed to get the two rule sets used to classify the gender of the subject for every row in the test dataset.

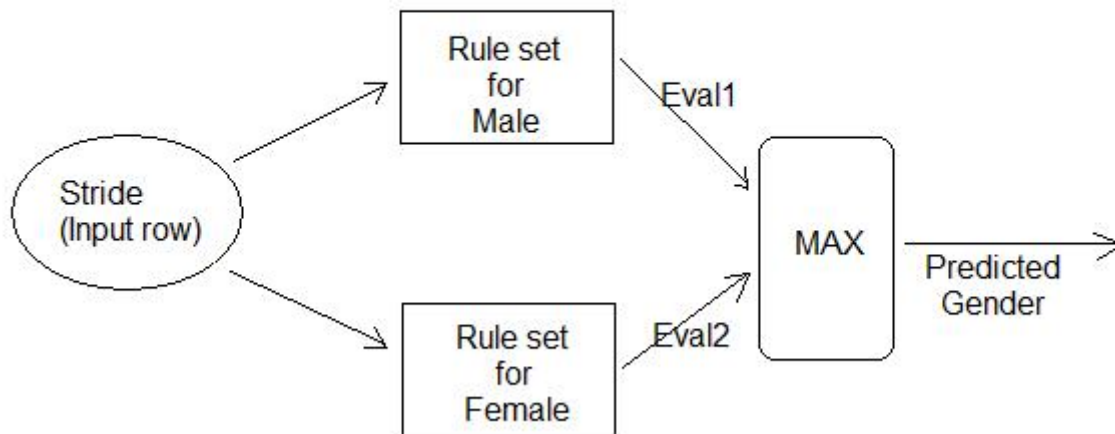


Figure 4.23 – Classifying the gender of a subject based on an input stride

As shown in Fig. 4.23, to predict the gender of a subject, the classification algorithm was run by inputting a row (representing a stride) from the test dataset to the rule sets discovered for both genders. The continuous values of the predicting attributes in the input row were fuzzified then each rule set evaluated the input row by counting the number of fuzzy rules in the set that

were fully matching the input fuzzified row. If the value of Eval1 is greater than the value of Eval2, the classification algorithm would predict the gender to be male; otherwise the input stride would be predicted to come from a female subject. Different sizes were tested for both rule sets in Fig. 4.23 in the same manner discussed in section 4-3. The tested sizes for the rule sets were 10, 20, 30, and 40 rules per set. Fig. 4.24 shows the classification of the gender based on an input experiment (a set of 120 strides).

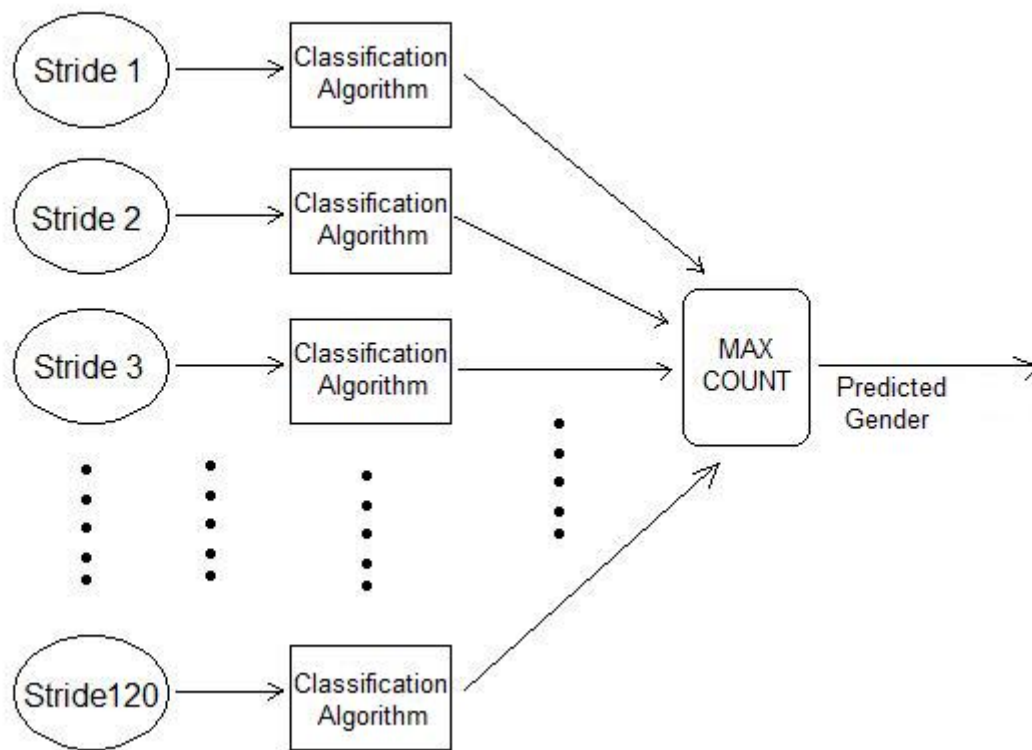


Figure 4.24 – Classifying the gender of a subject based on an experiment

The resultant rule sets from the second and third GA variants were merged to study the effect of combining rules from different methods on the hit ratio of the classification algorithm, two sizes of the combined rule sets were studied: 10 and 20 rules per rule set.

4-6 BMI Classification Experiments

The objective of the experiments discussed in this section is to test if the proposed fuzzy genetic algorithm can discover classification rules that can classify the BMI of a subject given the values of predicting attributes of one stride or a set of strides (experiment). The predicting attributes were all the attributes mentioned in section 4-1 except for the BMI which is the goal attribute. The dataset used for this task was taken from all the twenty-three subjects. As was shown in Table 4.1, the data for thirteen subjects formed the training dataset while the data from the remaining ten subjects formed the test dataset. Table 4.4 shows the properties of the subjects in both datasets.

A procedure very similar to the one discussed in sections 4-3 and 4-5 was followed. The three variants of the proposed GA were experimented to study the effect of the interestingness measure. All the 120 strides of the experiments of subjects in the training set were used as input to the GAs. The size of the training dataset was 4440 rows (strides) and 15 columns (attributes). The size of the test dataset was 2880 rows and 15 attributes.

A separate GA run was used to discover the fuzzy classification rule sets for the three values defined for the BMI (underweight, natural, overweight). This means for every variant of the proposed GA, three runs were executed to get the three rule sets used to classify the BMI of the subject for every row in the test dataset.

As shown in Fig. 4.25, to predict the BMI of a subject, the classification algorithm was run by inputting a row (representing a stride) from the test dataset to the rule sets discovered for each of the three BMI output values. The continuous values of the predicting attributes in the input

row were fuzzified then each rule set evaluated the input row by counting the number of fuzzy rules in the set that were fully matching the input fuzzified row. If the value of Eval1 is greater than the value of Eval2 and Eval3, the classification algorithm would predict the BMI to be underweight; if Eval2 is the maximum, the input stride would be predicted to come from a subject having natural BMI; otherwise the BMI would be predicted to come from an overweight subject. Different sizes were tested for both rule sets in Fig. 4.25 in the same manner discussed in sections 4-3 and 4-5. The tested sizes for the rule sets were 10, 20, 30, and 40 rules per set. Fig. 4.26 shows the classification of the BMI based on an input experiment (a set of 120 strides).

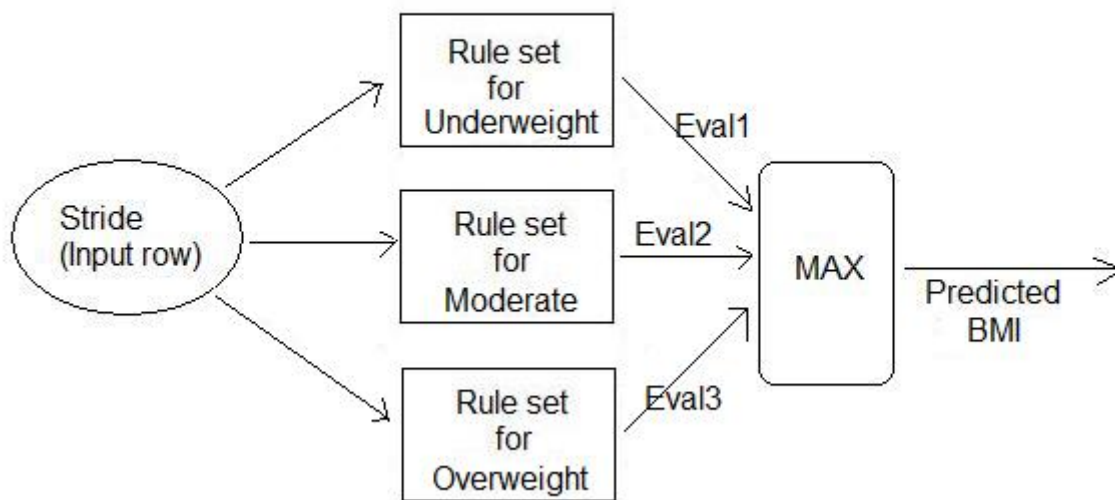


Figure 4.25 – Classifying the BMI of a subject based on an input stride

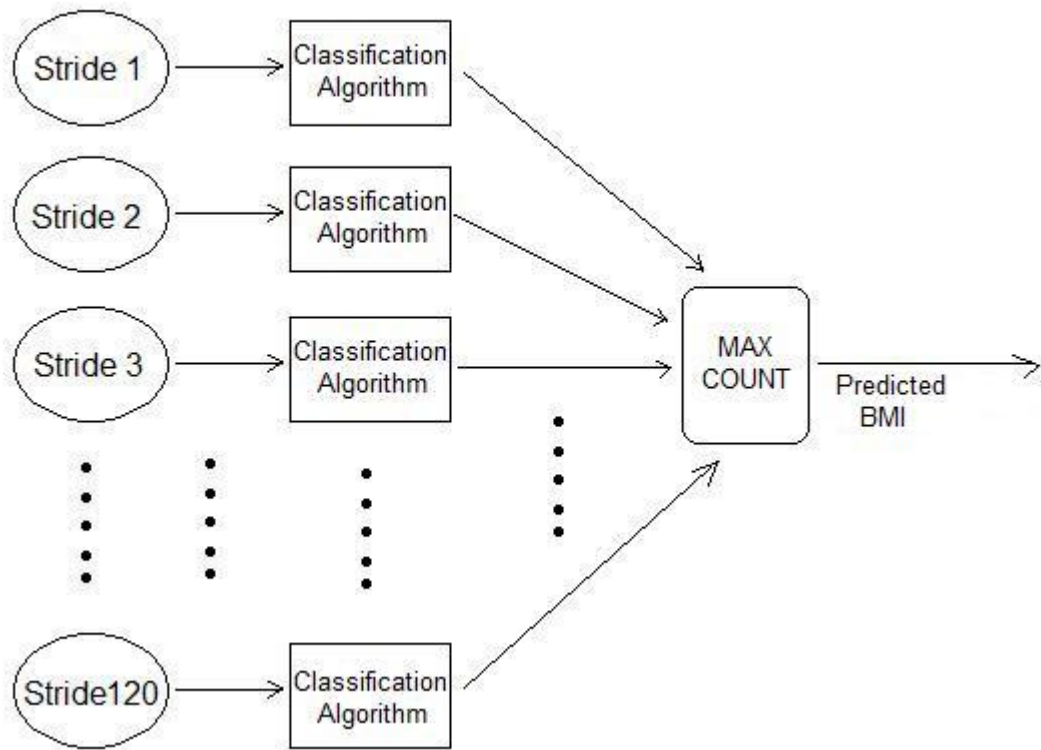


Figure 4.26 – Classifying the BMI of a subject based on an experiment

The resultant rule sets from the second and third GA variants were merged to study the effect of combining rules from different methods on the hit ratio of the classification algorithm, two sizes of the combined rule sets were studied; 10 and 20 rules per rule set.

4-7 Speed Classification Experiments

The objective of the experiments discussed in this section is to test if the proposed fuzzy genetic algorithm can discover classification rules that can classify the speed (slow, normal, fast) of a subject given the values of predicting attributes of one stride or a set of strides (experiment). The predicting attributes were all the attributes mentioned in section 4-1 except for the speed of the experiments which is the goal attribute. The dataset used for this task was taken from all the twenty-three subjects. As was shown in Table 4.1, the data for thirteen subjects formed the training dataset while the data from the remaining ten subjects formed the test dataset. Table 4.4 shows the properties of the subjects in both datasets.

A procedure very similar to the one discussed in sections 4-3 and 4-5 was followed. The three variants of the proposed GA were experimented to study the effect of the interestingness measure. All the 120 strides of the experiments of subjects in the training set were used as input to the GAs. The size of the training dataset was 4440 rows (strides) and 15 columns (attributes). The size of the test dataset was 2880 rows and 15 attributes.

A separate GA run was used to discover the fuzzy classification rule sets for the three values defined for the speed of the experiment (slow, natural, fast). This means for every variant of the proposed GA, three runs were executed to get the three rule sets used to classify the speed of the subject for every row in the test dataset.

As shown in Fig. 4.27, to predict the speed of a subject, the classification algorithm was run by inputting a row (representing a stride) from the test dataset to the rule sets discovered for each of the three speed output values. The continuous values of the predicting attributes in the input

row were fuzzified then each rule set evaluated the input row by counting the number of fuzzy rules in the set that were fully matching the input fuzzified row. If the value of Eval1 is greater than the value of Eval2 and Eval3, the classification algorithm would predict the speed to be slow; if Eval2 is the maximum, the input stride would be predicted to come from a subject walking at his/her natural speed; otherwise the speed would be predicted to come from a fast walking experiments. Different sizes were tested for both rule sets in Fig. 4.27 in the same manner discussed in sections 4-3 and 4-5. The tested sizes for the rule sets were 10, 20, 30, and 40 rules per set. Fig. 4.28 shows the classification of the speed based on an input experiment (a set of 120 strides).

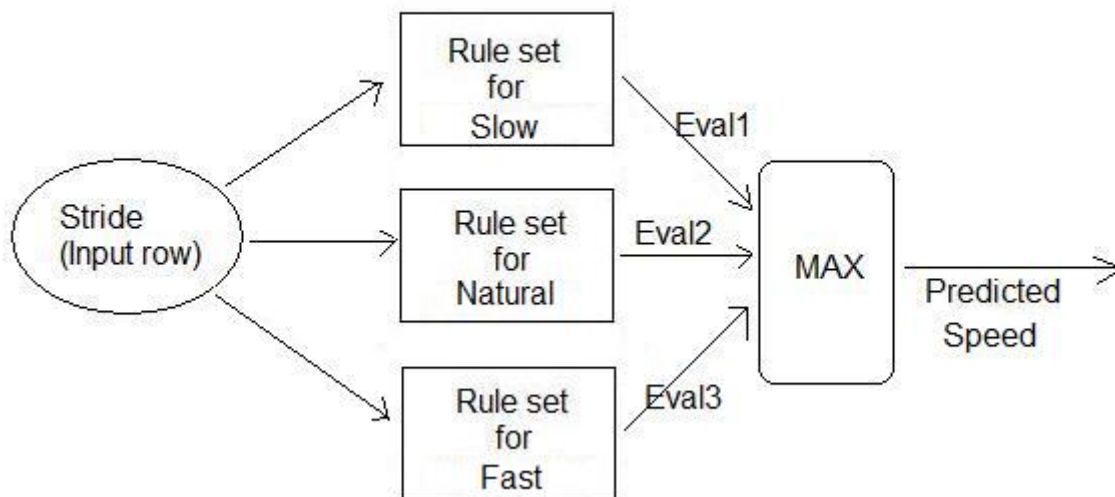


Figure 4.27 – Classifying the speed of a subject based on an input stride

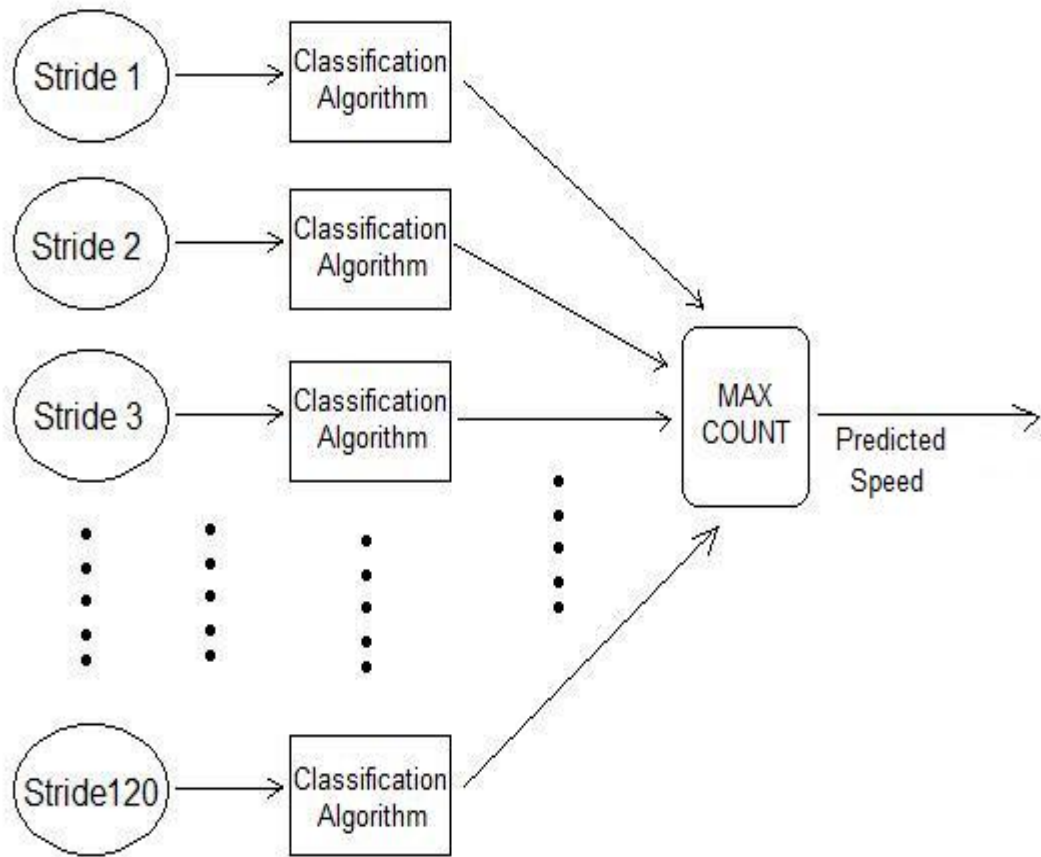


Figure 4.28 – Classifying the speed of a subject based on an experiment

The resultant rule sets from the second and third GA variants were merged to study the effect of combining rules from different methods on the hit ratio of the classification algorithm, two sizes of the combined rule sets were studied; 10 and 20 rules per rule set.

4-8 Age Classification Experiments

The objective of the experiments discussed in this section is to test if the proposed fuzzy genetic algorithm can discover classification rules that can classify the age group of a subject given the values of predicting attributes of one stride or a set of strides (experiment). The predicting attributes were all the attributes mentioned in section 4-1 except for the age group which is the goal attribute. The dataset used for this task was taken from all the twenty-three subjects. As was shown in Table 4.1, the data for thirteen subjects formed the training dataset while the data from the remaining ten subjects formed the test dataset. Table 4.4 shows the properties of the subjects in both datasets.

A procedure very similar to the one discussed in sections 4-3 and 4-5 was followed. The three variants of the proposed GA were experimented to study the effect of the interestingness measure. All the 120 strides of the experiments of subjects in the training set were used as input to the GAs. The size of the training dataset was 4440 rows (strides) and 15 columns (attributes). The size of the test dataset was 2880 rows and 15 attributes.

A separate GA run was used to discover the fuzzy classification rule sets for the three defined age groups. This means for every variant of the proposed GA, three runs were executed to get the three rule sets used to classify the age group of the subject for every row in the test dataset.

As shown in Fig. 4.29, to predict the speed of a subject, the classification algorithm was run by inputting a row (representing a stride) from the test dataset to the rule sets discovered for each of the three age group output values. The continuous values of the predicting attributes in the

input row were fuzzified then each rule set evaluated the input row by counting the number of fuzzy rules in the set that were fully matching the input fuzzified row. If the value of Eval1 is greater than the value of Eval2 and Eval3, the classification algorithm would predict the age to be less than 24 years old (age group 1); if Eval2 is the maximum, the input stride would be predicted to come from a subject of age between 24 and 35 years old (age group 2); otherwise the age would be predicted to be 36 years old or more (age group 3). Different sizes were tested for both rule sets in Fig. 4.29 in the same manner discussed in sections 4-3 and 4-5. The tested sizes for the rule sets were 10, 20, 30, and 40 rules per set. Fig. 4.30 shows the classification of the speed based on an input experiment (a set of 120 strides).

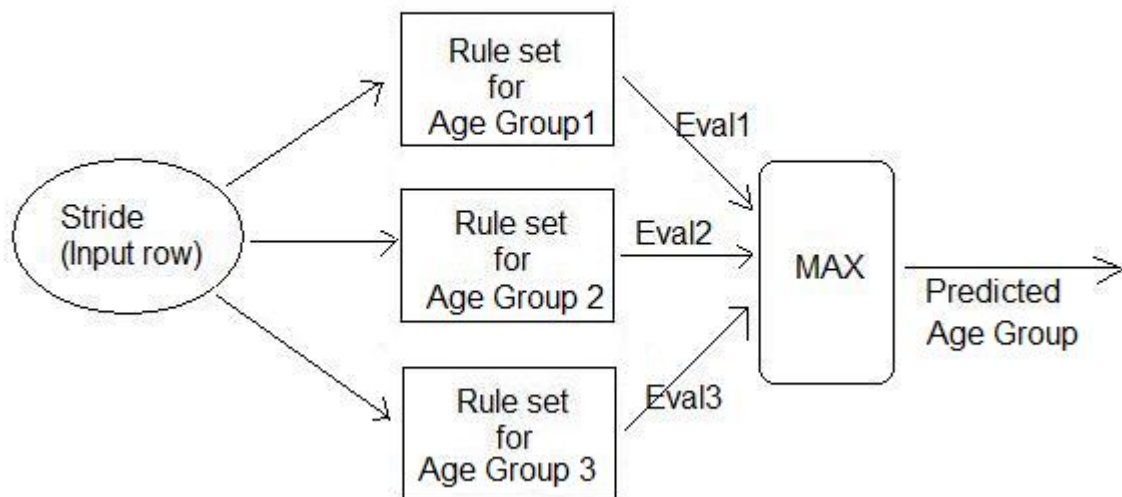


Figure 4.29 – Classifying the age group of a subject based on an input stride

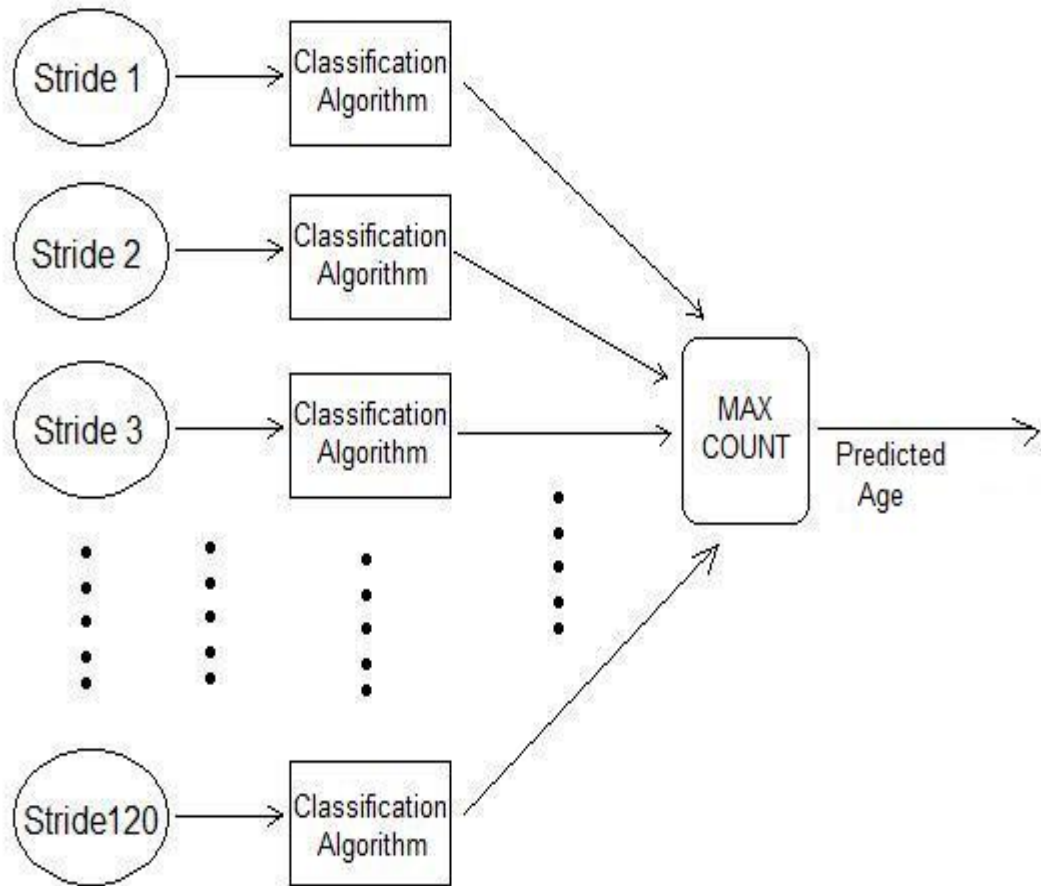


Figure 4.30 – Classifying the age group of a subject based on an experiment

The resultant rule sets from the second and third GA variants were merged to study the effect of combining rules from different methods on the hit ratio of the classification algorithm, two sizes of the combined rule sets were studied; 10 and 20 rules per rule set.

CHAPTER 5

Results and Interpretation

5-1 Results of Gait Identification Experiments

The results for running the identification algorithm with the first variant of the GA (interestingness of rules was not considered) at the four sizes of the rule sets are summarized in Table 5.1.

Table 5.1 - Results for the GA variant with accuracy only (Gait Identification)

Rules per Set	Per Stride (1480)			Per Experiment (37)		
	Hits	Misses	Hit %	Hits	Misses	Hit %
10 Rules	947	533	64%	24	13	64.9%
20 Rules	1017	463	68.7%	26	11	70.3%
30 Rules	1057	423	71.4%	27	10	73%
40 Rules	1062	418	71.8%	27	10	73%

Table 5.1 shows that the best hit rate for the identification algorithm without considering the interestingness measure in the GA fitness function was limited to 73% (per experiment) and 71.8% (per stride). It is clear from the table that the hit rate increases slightly by increasing the size of the rule set so the prediction power is not concentrated in few fuzzy rules but we might need the cooperation of a large number of rules to achieve a higher hit rate.

It can also be seen from Table 5.1 that the hit rate increased very slightly when a whole experiment was considered (40 strides), which means that providing more than one stride for a subject did not have a significant role in increasing the identification hit rate.

The results for running the identification algorithm with the second variant of the GA (fixed weights) at the four sizes of the rule sets are summarized in table 5.2.

Table 5.2 - Results for the GA variant with fixed weights (Gait Identification)

Rules per Set	Per Stride (1480)			Per Experiment (37)		
	Hits	Misses	Hit %	Hits	Misses	Hit %
10 Rules	1152	328	77.8%	32	5	86.5%
20 Rules	1289	191	87.1%	33	4	89.2%
30 Rules	1329	151	89.8%	34	3	91.9%
40 Rules	1342	138	90.7%	34	3	91.9%

Table 5.2 shows the significant increase in the identification hit rate (around 20%) when the interestingness measure was added to the fitness function of the GA. Yet, it is still clear from the table that the hit rate increases by increasing the size of the rule set so the prediction power is still not concentrated in few fuzzy rules but we might need the cooperation of a large number of rules to achieve a higher hit rate.

It can also be seen from Table 5.2 that the hit rate increased slightly when a whole experiment was considered (40 strides), so providing more than one stride for a subject did not have a significant role in increasing the identification hit rate.

The results for running the identification algorithm with the third variant of the GA (adapting weights) at the four sizes of the rule sets are summarized in table 5.3.

Table 5.3 - Results for the GA variant with adapting weights (Gait Identification)

Rules per Set	Per Stride (1480)			Per Experiment (37)		
	Hits	Misses	Hit %	Hits	Misses	Hit %
10 Rules	1240	240	83.8%	31	6	83.8%
20 Rules	1240	240	83.8%	31	6	83.8%
30 Rules	1240	240	83.8%	31	6	83.8%
40 Rules	1240	240	83.8%	31	6	83.8%

It can be seen from Table 5.3 that the identification hit rate was not as good as when the weights for the measures of interestingness and predictive accuracy were fixed. It is still much better than when no interestingness measure was considered. It is also clear from the table that the hit rate is the same for all experimented sizes of the rule set, so here it can be said that the prediction power is concentrated in few fuzzy rules, adding more identification rules has no effect on the hit rate.

It can also be seen from Table 5.3 that the hit rate did not change when a whole experiment is considered (40 strides), so providing one stride for a subject was enough to achieve the best identification hit rate for this method.

The results for running the identification algorithm with the combined rule sets from second and third variants of the GA (fixed and adapting weights) at two sizes of the rule sets are summarized in table 5.4.

Table 5.4 - Results for combining fixed and adapting weights (Gait Identification)

Rules per Set	Per Stride (1480)			Per Experiment (37)		
	Hits	Misses	Hit %	Hits	Misses	Hit %
10 Rules	1380	100	93.2%	35	2	94.6%
20 Rules	1380	100	93.2%	35	2	94.6%

By studying the results achieved in Table 5.4, it can be said that combining the two GAs with fixed and adapting weights for the measures of the interestingness and predictive accuracy did combine the advantages of both algorithms. This method achieved the best identification hit rate (even higher than the fixed weights algorithm). Moreover, the highest hit rate was achieved at the smallest size of the rules sets (10 rules), thus the identification power is concentrated in few rules per rule set, which was the same advantage for the adapting weights algorithm.

Comparing the hit rates when a single stride was considered to when a whole experiment was considered (40 strides), the hit rate very slightly increased (1.4%) meaning that it was almost enough to feed the identification algorithm with the data from a single stride of a subject.

Following are the best rules obtained for each subject using the fixed weights GA:

If BMI is Moderate & Stride time is Average & Stride length is Average & Stance/Swing ratio is

Average & Double Stance phase ratio is Small & Soleus peak shift is Right

& Tibialis peak shift is Average Then Subject = M1

If Stride length is Large & Double Stance phase ratio is Average & LFP is Average

& Soleus peak shift is Left & Tibialis peak shift is Left

& Gastrocnemius peak shift is Left Then Subject = M2

If Stride time is Large & Stride width is Small & Stance/Swing ratio is Average

& Double Stance phase ratio is Average & LFP is Average & Tibialis peak shift is Left

& Gastrocnemius peak shift is Left & Vastus peak shift is Average

Then Subject = M3

If Stride time is Large & Stance/Swing ratio is Large & Double Stance phase ratio is Average

& LFP is Large & Tibialis peak shift is Average & Vastus peak shift is Left

Then Subject = M4

If BMI is Moderate & Stride time is Average & Stride width is Large & Stance/Swing ratio is

Average & Double Stance phase ratio is Average & Tibialis peak shift is Left

& Gastrocnemius peak shift is Left Then Subject = M5

If Stride time is Large & Stride length is Large & Double Stance phase ratio is Average
& Tibialis peak shift is Right Then Subject = M6

If Stride time is Average & Stride length is Large & Stance/Swing ratio is Large
& Double Stance phase ratio is Large & LFP is Small & Soleus peak shift is Right
& Tibialis peak shift is Average & Gastrocnemius peak shift is Average
Then Subject = F1

If BMI is Underweight & Stride time is Average & Stride width is Small
& Double Stance phase is Average & Tibialis peak shift is Average
Then Subject = F2

If Stride time is Average & Stride length is Large & Stance/Swing ratio is Average
& Double Stance phase ratio is Average & Soleus peak shift is Right & Tibialis peak shift is Left
& Vastus peak shift is Right Then Subject = F3

If Height is Short & Stride time is Average & Stride width is Small
& Stance/Swing ratio is Average & Double Stance phase ratio is Average
& Gastrocnemius peak shift is Average & Vastus peak shift is Left
Then Subject = F4

If BMI is Underweight & Stride time is Average & Stance/Swing ratio is Average

& Double Stance phase ratio is Average & Soleus peak shift is Average

Then Subject = F5

If Stride time is Small & Stride length is Small & Stride width is Small

& Stance/Swing ratio is Large & Double Stance phase ratio is Large

& Soleus peak shift is Left & Tibialis peak shift is Left

Then Subject = F6

If BMI is Natural & Stride time is Large & Stride width is Small

& Stance/Swing ratio is Large & Double Stance phase ratio is Large

& Soleus peak shift is Left & Tibialis peak shift is Left

& Vastus peak shift is Average Then Subject = F7

5-2 Results of Gait Similarity Experiments

The results of running the 24 experiments for testing the gait similarity algorithm are summarized in Tables 5.5, 5.6, and 5.7 for fixed weights GA, adapting weights GA, and combined fixed/adapting weights system respectively. The first column in each table shows the input experiment (120 strides) from the test dataset. Beside the subject ID, a letter is used to show whether this experiment was a natural walking, slow walking, or fast walking (N for natural, F for fast, and S for slow).

The subject columns in the three tables show for each experiment which subject from the training dataset was predicted by the three algorithms to be similar to the input test subject. The similarity columns show how close the input and output subjects are based on the simple similarity measure discussed in section 4-4. The rank columns show if the predicted subject is the closest to the input subject among the thirteen subjects in the training dataset or how many subjects are closer to the input subject. For each test subject, the subjects in the training dataset were ranked from 1 (most similar) to 13 (least similar) to the test subject based on the similarity matrix that was previously shown in table 4.3.

Some promising points that can be noticed from Table 5.5 (GA with fixed weights) are:

- 1- For all experiments, the mapped subject was at least 50% similar to the input subject.
- 2- In 62.5% of the experiments, the GA with fixed weights was able to map the most similar subject in the training set to the subject of the input experiment. For 12.5% of the experiments, the mapped subject was the second closest to the input subject, for 16.67% of the experiments, the mapped subject was the fourth closest to the input subject, and for

the remaining experiments (8.3%) the mapped subject was the fifth closest to the input subject.

- 3- For natural walking experiments, 80% of the experiments resulted in mapping the most similar subject to the input subject, for fast walking experiments, 57% of the experiments were mapped to the most similar subject in the input experiment, whereas for slow walking experiments, only 43% of the experiments were mapped to the most similar subject in the input experiment.
- 4- For all experiments, the mapped subject was of the same gender of the input subject.

Some promising points that can be noticed from Table 5.6 (GA with adapting weights) are:

- 1- For all experiments, the mapped subject was at least 50% similar to the input subject.
- 2- In 58.3% of the experiments, the GA with adapting weights was able to map the most similar subject in the training set to the subject of the input experiment. For 12.5% of the experiments, the mapped subject was the second closest to the input subject, for 12.5% of the experiments, the mapped subject was the third closest to the input subject, and for 16.67% of the experiments, the mapped subject was the fourth closest to the input subject.
- 3- For natural walking experiments, 60% of the experiments resulted in mapping the most similar subject to the input subject, for fast walking experiments, 57% of the experiments were mapped to the most similar subject in the input experiment, whereas for slow walking experiments, only 57% of the experiments were mapped to the most similar subject in the input experiment.
- 4- For all experiments, the mapped subject was of the same gender of the input subject.

Some promising points that can be noticed from Table 5.7 (Combined Algorithm) are:

- 1- For all experiments, the mapped subject was at least 50% similar to the input subject.
- 2- In 70.8% of the experiments, the similarity algorithm was able to map the most similar subject in the training set to the subject of the input experiment. For 12.5% of the experiments, the mapped subject was the second closest to the input subject, for 4% of the experiments, the mapped subject was the third closest to the input subject, and for 12.5% of the experiments, the mapped subject was the fourth closest to the input subject.
- 3- For natural walking experiments, 80% of the experiments resulted in mapping the most similar subject to the input subject, for fast walking experiments, 71.4% of the experiments were mapped to the most similar subject in the input experiment, whereas for slow walking experiments, only 57% of the experiments were mapped to the most similar subject in the input experiment.
- 4- For all experiments, the mapped subject was of the same gender of the input subject.

The results for the three algorithms are summarized in Table 5.8 and Table 5.9.

Table 5.5 - Results for gait similarity experiments (Fixed weights)

Experiment	Fixed weights (40 rules)		
	Subject	Similarity	Rank
M7 N	M1	64.3%	1
M8 N	M4	71.4%	1
M9 N	M6	57.1%	4
M10 N	M5	64.3%	1
M11 N	M1	85.7%	1
F8 N	F6	57.1%	1
F9 N	F4	64.3%	1
F10 N	F2	64.3%	1
F11 N	F1	50%	1
F12 N	F2	57.1%	4
M8 F	M2	64.3%	2
M10 F	M2	50%	5
M11 F	M1	85.7%	1
F8 F	F6	57.1%	1
F9 F	F4	64.3%	1
F10 F	F2	64.3%	1
F12 F	F2	57.1%	4
M8 S	M2	64.3%	2
M10 S	M3	50%	5
M11 S	M1	85.7%	1
F8 S	F6	57.1%	1
F9 S	F4	64.3%	1
F10 S	F4	50%	4
F12 S	F7	64.3%	2

Table 5.6 - Results for gait similarity experiments (Adapting weights)

Experiment	Adapting weights (40 rules)		
	Subject	Similarity	Rank
M7 N	M1	64.3%	1
M8 N	M4	71.4%	1
M9 N	M3	57.1%	4
M10 N	M5	64.3%	1
M11 N	M1	85.7%	1
F8 N	F5	50%	3
F9 N	F4	64.3%	1
F10 N	F7	50%	4
F11 N	F1	50%	1
F12 N	F2	57.1%	4
M8 F	M2	64.3%	2
M10 F	M6	64.3%	1
M11 F	M1	85.7%	1
F8 F	F4	57.1%	1
F9 F	F4	64.3%	1
F10 F	F6	57.1%	3
F12 F	F2	57.1%	4
M8 S	M2	64.3%	2
M10 S	M6	64.3%	1
M11 S	M1	85.7%	1
F8 S	F4	57.1%	1
F9 S	F4	64.3%	1
F10 S	F6	57.1%	3
F12 S	F7	64.3%	2

Table 5.7 - Results for gait similarity experiments (Combined fixed/adapting weights)

Experiment	Combined (20 rules)		
	Subject	Similarity	Rank
M7 N	M1	64.3%	1
M8 N	M4	71.4%	1
M9 N	M3	57.1%	4
M10 N	M5	64.3%	1
M11 N	M1	85.7%	1
F8 N	F6	57.1%	1
F9 N	F4	64.3%	1
F10 N	F2	64.3%	1
F11 N	F1	50%	1
F12 N	F2	57.1%	4
M8 F	M2	64.3%	2
M10 F	M6	64.3%	1
M11 F	M1	85.7%	1
F8 F	F6	57.1%	1
F9 F	F4	64.3%	1
F10 F	F2	64.3%	1
F12 F	F2	57.1%	4
M8 S	M2	64.3%	2
M10 S	M6	64.3%	1
M11 S	M1	85.7%	1
F8 S	F6	57.1%	1
F9 S	F4	64.3%	1
F10 S	F6	57.1%	3
F12 S	F7	64.3%	2

Table 5.8 – Summary of results for gait similarity experiments (all experiments)

Mapped Subject	Fixed weights	Adapting weights	Combined System
Most similar	62.5%	58.3%	70.8%
2 nd most similar	12.5%	12.5%	12.5%
3 rd most similar	0%	12.5%	4%
4 th most similar	16.67%	16.67%	12.5%
5 th most similar	8.3%	0%	0%

Table 5.9 – Summary of results for gait similarity experiments grouped by speed

Mapped Subject	Fixed weights	Adapting weights	Combined System
Natural speed experiments	80%	60%	80%
Fast speed experiments	57%	57%	71.4%
Slow speed experiments	43%	57%	57%

5-3 Results of Gait Classification Experiments

The results of running the gender classification experiments described in section 4-5 did not achieve a good classification hit rate. For the first GA variant, the best hit rate was less than 50%. Tables 5.10 through 5.12 summarize the results for running the fixed weights GA, the adapting weights GA, and the combined system.

Table 5.10 - Results for the GA variant with fixed weights (Gender Classification)

Rules per Set	Per Stride (2880)			Per Experiment (24)		
	Hits	Misses	Hit %	Hits	Misses	Hit %
10 Rules	2061	819	71.6%	18	6	75%
20 Rules	2109	771	73.2%	18	6	75%
30 Rules	2182	698	75.8%	19	5	79.2%
40 Rules	2204	676	76.5%	19	5	79.2%

Table 5.11 - Results for the GA variant with adapting weights (Gender Classification)

Rules per Set	Per Stride (2880)			Per Experiment (24)		
	Hits	Misses	Hit %	Hits	Misses	Hit %
10 Rules	2166	714	75.2%	19	5	79.2%
20 Rules	2174	706	75.5%	19	5	79.2%
30 Rules	2174	706	75.5%	19	5	79.2%
40 Rules	2181	699	75.7%	19	5	79.2%

Table 5.12 - Results for combining fixed and adapting weights (Gender Classification)

Rules per Set	Per Stride (2880)			Per Experiment (24)		
	Hits	Misses	Hit %	Hits	Misses	Hit %
10 Rules	2223	657	77.2%	19	5	79.2%
20 Rules	2223	657	77.2%	19	5	79.2%

CHAPTER 6

Conclusions and Future Considerations

6-1 Conclusions

- 1- The effect of considering a measure for the rule interestingness beside the predictive accuracy measure was significant in increasing the classification hit rates for all experiments that were run in this study.
- 2- For the gait identification experiments, the proposed fuzzy genetic algorithm achieved high identification hit rates.
- 3- The rule sets discovered by fixed weights genetic algorithm were capable of achieving higher classification hit rates than the rule sets discovered by the adapting weights genetic algorithm.
- 4- When using fixed weights for interestingness and predictive accuracy, the classification hit rates tended to increase as the number of rules considered per rule set increased, the classification power for this technique was distributed among the whole rule set.
- 5- When using adapting weights for interestingness and predictive accuracy, the classification hit rates remained constant or slightly increased as the number of rules considered per rule set increased, the prediction power was concentrated in a small number of rules (the best rules in the rule set).

- 6- Combining the rule sets from the fixed weights and adapting weights methods resulted in achieving the best classification hit rates (up to 94.6%) using the least number of rules (10 per rule set), thus achieving the advantages of both methods.
- 7- The previous point supports the idea that for classification task (and for data mining tasks in general), it is not enough to compare the rule sets resulting from different techniques and algorithms and determine which is better. It is also important to test if rule sets resulting from different techniques can be merged somehow to achieve better results.
- 8- The genetic algorithm with fixed weights was successful in the gait similarity task. In all experiments, it was able to map the strides of an input unknown subject to a known subject of the same gender and with a similarity of at least 50% based on the similarity measure defined in this study.
- 9- For the gait similarity experiments, the genetic algorithm was noticeably more successful when the unknown subject is walking at natural speed than when he/she walks at a slower or a faster speed. Practically speaking, if an unknown impaired subject is to be mapped to a known subject with similar gait, it is more likely that he/she will be asked to walk at his/her natural speed.
- 10- The proposed fuzzy genetic algorithm did not achieve high classification hit rates for classifications based on gender, BMI, speed, and age group. This can be attributed to the small sizes of the training and test datasets, the classification rules might be skewed by the specific nature of the participating subjects in the training dataset, and thus not be general enough to successfully work with the characteristics of the new subjects in the test dataset.

Also, the use of the same predicting attributes with different goal attributes might not be the best choice. It might be required to carefully choose which predicting attributes are more appropriate for each goal attribute, which features to extract for EMG muscles for instance, and the number and boundaries of the defined fuzzy membership functions for the chosen continuous predicting attributes.

11- The execution times for all runs of different variants of the genetic algorithms which are sequential are considerably long. On a Pentium 4 laptop with 512 MB Ram, each run needed between 20 and 30 minutes. This is because for every tested rule in every generation, the whole training dataset should be scanned which consumes considerable time. It is very desirable to study ways to optimize the simple genetic algorithm, specially through parallelization.

6-2 Future Considerations

The following are some considerations to build on what was achieved through this study:

- 1- The proposed fuzzy genetic algorithm need to be applied on a larger dataset composed of both healthy and impaired subjects. For the laboratory of human motion analysis and rehabilitation in UTEP, it is planned to build a dataset for the gait of at least 200 healthy subjects and as many impaired subjects as possible.
- 2- More attributes can be considered like the subjects' joint angles, and more EMG muscles and features can be used for the various classification tasks studied here.
- 3- It might also be useful to compare (and merge if possible) the results of the proposed genetic algorithm with other techniques for classification in literature that might be based on evolutionary computation or any other techniques.
- 4- It is also useful to try to incorporate subjective measures for interestingness of discovered rules to make use of the feedback of available domain experts.
- 5- It is important with a bigger dataset to study ways to parallelize the proposed genetic algorithm for more acceptable run times.

References

- [1] A.A. Freitas. “A survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery,” in *Advances in Evolutionary Computing: Theory and Applications*. A. Ghosh and S. Tsutsui, Ed. Springer-Verlag, 2003, pp. 819-845.
- [2] J. Anderson and A. Kotsiopoulos. “Enhanced Decision Making using Data Mining: Applications for Retailers.” *Journal of textile and apparel, Technology and Management JTATM*, Vol. 2, Issue 3, Summer 2002.
- [3] S. Sumathi and S.N. Sivanandam. *Introduction to Data Mining and its Applications*. Springer, 2006.
- [4] Lipo Wang and Xiuju Fu. *Data Mining with Computational Intelligence*. Springer, 2005.
- [5] E. Noda and A.A. Freitas. “Discovering Knowledge Nuggets with a Genetic Algorithm,” in *Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques*. Springer, 2006, pp. 395-432.
- [6] I. Ntoutsi, N. Pelekis, Y. Theodoridis. “Pattern Comparison in Data Mining: A Survey,” in *Research and Trends in Data Mining Technologies and Applications*. David Tanar, Ed. Idea Group Publishing, 2007, pp. 86-121.

- [7] C.M. Hill, L.C. Malone, and L. Trocine. “Data Mining and Traditional Regression,” in *Statistical Data Mining and Knowledge Discovery*. H. Bozdogan, Ed. Chapman & Hall/CRC, 2004, pp. 233-246.
- [8] M.H. Dunham. *Data Mining – Introductory and Advanced Topics*. Prentice Hall, 2003.
- [9] B. Padmanabhan and A. Tuzhilin. “Unexpectedness as a measure of interestingness in knowledge discovery.” *Decision Support Systems* 27, 1999, pp. 303-318.
- [10] G.M. Marakas. *Modern Data Warehousing, Mining, and Visualization – Core Concepts*. Pearson Education, Inc, 2003.
- [11] R.J. Hilderman and H.J. Hamilton. “Applying Objective Interestingness Measures in Data Mining Systems,” in *Proc. of the 4th European Symposium on Principles of Data Mining and Knowledge Discovery PKDD’00*, pages 432-439, Sep. 2000.
- [12] C. Giannella and E. Robertson. “On approximation measures for functional dependencies.” *Information Systems* 29: 483-507, 2004.
- [13] A.A. Freitas. “On Rule Interestingness Measures.” *Knowledge-Based Systems journal*, Vol. 12 (5-6):309-315, Oct. 1999.

- [14] R.J. Hilderman and H.J. Hamilton. “Knowledge Discovery and Interestingness Measures: A Survey.” *Department of Computer Science, University of Regina, Saskatchewan, Canada, Technical Report CS 99-04*, Oct. 1999.
- [15] Andrea Tettamanzi and Marco Tomassini. *Soft Computing, Integrated Evolutionary, Neural, and Fuzzy Systems*. Springer, 2001.
- [16] L. Fortuna, G. Rizzotto, M. Lavorgna, G. Nunnari, M. G. Xibilia, and R. Caponetto. *Soft Computing*. Springer, 2001.
- [17] Z. Michalewicz, R. HinterDing, M. Michalewicz. “Evolutionary Algorithms,” in *Fuzzy Evolutionary Computation*. Witold Pedrycz, Ed. Kluwer Academic Publishers, 1997, pp. 3-23.
- [18] Keneth De Jong. “Two Grand Challenges for EC, Unification and Expansion,” in *Frontiers of Evolutionary Computation*. Anil Menon, Ed. Kluwer Academic Publishers, 2004, pp. 37-47.
- [19] R.G.S. Asthana. “Evolutionary Algorithms and Neural Networks,” in *Soft Computing and Intelligent Systems Theory and Applications*. N.K. Sinha and M.M. Gupta, Ed. Academic Press, 2000, pp. 111-136.
- [20] E. Noda, A.A. Freitas, and H.S. Lopes. “Discovering Interesting Prediction Rules with a Genetic Algorithm.” in *Proc. of the Congress on Evolutionary Computation CEC'99*, Vol 2: 1322-1329, IEEE Press, 1999.

- [21] W. Romao, A.A. Freitas, and I.M.S. Gimenes. “Discovering interesting knowledge from a science and technology database with a genetic algorithm.” *Applied Soft Computing* 4: 121-137, 2004.
- [22] I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1st edition, 1999.
- [23] K.R. Kaufmann and D.H. Sutherland. “Kinematics of Normal Human Walking,” in *Human Walking, Third Edition*. Jessica Rose and James G. Gamble, Ed. Lippincott Williams & Wilkins, 2006, pp. 33-51.
- [24] C.L. Vaughan, B.L. Davis, J.C. O’connor. *Dynamics of Human Gait*. Human Kinetics Publishers, 1992.

Curriculum Vitae

Abdallah Abdel-Rahman Hassan Mahmoud was born in April 1977 in Cairo, Egypt. He got his bachelor's degree in Computer Engineering from Cairo University, Egypt. For three years, he worked as a software engineer developing systems using C++ mainly for a number of companies in Cairo, Egypt. During that time he got MCAD certificate (Microsoft Certified Application Developer). In June 2003, he decided to live in a small city in south of Egypt; Minia, where he started working as a teaching assistant in the Department of Computer and System Engineering at Minia University. He was awarded a Fulbright graduate scholarship to study in the USA, so he joined the University of Texas at El Paso in the fall semester of 2007. He is married and a father of a one-years-old girl. He has ambitious plans for his future researcher and university professor. His main interests are in studying artificial intelligence, soft computing, and knowledge engineering. He is a religious muslim and he will always be proud of that.

Permanent Address:

64 Nefertiti Street, Ard Sultan,

Minia, 61111

Egypt.

abdal_eg@yahoo.com

This thesis was typed by Abdallah Abdel-Rahman Hassan Mahmoud.