

Implementing a Fuzzy Classifier and Improving its Accuracy using Genetic Algorithms

Stig-Åke Svensson
Mälardalen University
M.Sc.Program in Computer
Engineering and Electronics
ssn03006@student.mdh.se

ABSTRACT

The use of fuzzy logic to describe abstract concepts and when designing decision making systems to make decisions that are much closer to the way a human does is an interesting and useful area to explore. To effectively implement these types of systems expert knowledge is required in the application area. This can be a problem if the people designing a system of this nature are not experts in the application area since it is not always possible to have an expert working continuously with a project. This problem can be alleviated by using data supplied by an expert to let the system learn from it and thereby improve the system without continuous input from experts.

In this paper the implementation of a fuzzy classifier that classifies the specie of the iris flower, based on the length and width of the sepal and petal, is presented. Its accuracy is then improved using a genetic algorithm to fine tune the membership functions.

Keywords

Fuzzy Sets, Fuzzy logic, Fuzzy Classifier, Genetic Algorithms

1. INTRODUCTION

In recent years the use of fuzzy logic in fuzzy systems has been implemented with good success in many different types of systems [8] ranging from controlling airplanes [7] to sake brewing [11]. As such, the knowledge on how to construct systems using fuzzy logic to solve problems is a valuable one.

This paper will describe an implementation of a fuzzy system that will be used to classify the specie of the iris flower based on the iris data set. This is a well known benchmark problem commonly used in machine learning research [1]. To improve the accuracy of the classifier a genetic algorithm (GA) is added to the implementation to fine tune the membership functions.

As will be seen the use of GA can be implemented with a noticeable improvement in the accuracy of the fuzzy classifier.

Much work has been done in the area of improving fuzzy classifiers [2, 3, 6, 12, 14]. Several different approaches have been taken such as generating the fuzzy rules in the classifier by using GA[2, 3, 6], by using GA to generate the whole classifier[14] or by using particle swarm optimization[12].

The paper is divided into sections as follows. Section 2 gives a brief explanation of some of the concepts regarding fuzzy sets and fuzzy logic. This background knowledge is extended on in section 3 where a general overview of a fuzzy system is given. Section 4 explains some basic concepts and the general idea behind GA. In section 5 the implementation is explained. Results are presented in section 6 and a conclusion is given in section 7. Acknowledgment is given in section 8, references in section 9 and an appendix containing raw data in section 10.

2. QUICK PRIMER ON FUZZY LOGIC

In this section a brief explanation of fuzzy sets and fuzzy logic will be given with the purpose of providing a basic knowledge of some of the methods applied later in the paper. A more in depth explanation can be found in [8] [13].

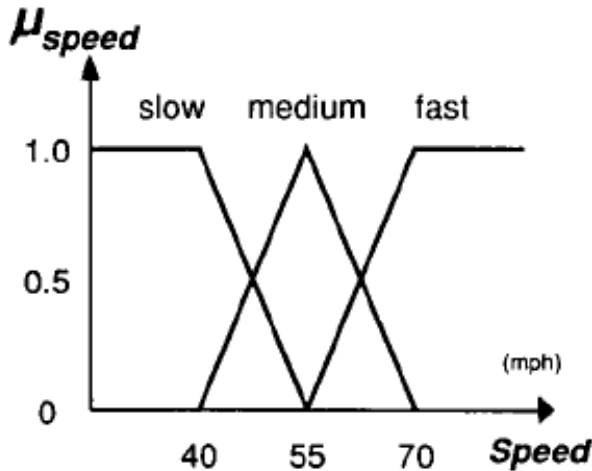
2.1 Fuzzy Sets

A collection of elements denoted by $\{u\}$ forms the universe U where u is a generic element of U . In classical set theory these elements may or may not belong to a particular crisp set. That is to say an element must either belong or not belong to the set. There is no possibility to partially belong to a set. In contrast, in fuzzy set theory, elements does not need to binary belong or not belong but can belong by a certain degree to a particular fuzzy set.

A fuzzy set F belonging to the universe U has a value x , in the interval $[0, 1]$, that corresponds to the element u . The value x represents the degree of membership which elements belong to F . This is defined as $\mu_F(u) = x$ where u is an element belonging to F and $\mu_F(u)$ is the membership function of F . In the case of when $\mu_F(u) = 0$ the element u does not belong to the set F . If $\mu_F(u) = 1$, u is considered to be fully part of F while if $0 < \mu_F(u) < 1$, u is considered to be a fuzzy member of F .

2.2 Fuzzy Set Operations

For this subsection A and B are fuzzy sets with the corresponding membership functions $\mu_A(u)$ and $\mu_B(u)$ in the universe U . The following definitions are given in [8].



[8]Figure 1: The membership functions of the fuzzy sets slow, medium and fast

2.2.1 Union

The union between two fuzzy sets(A and B) is described by the membership function $\mu_{A \cup B}(u)$ and is defined for all $u \in U$ by:

$$\mu_{A \cup B}(u) = \max(\mu_A(u), \mu_B(u))$$

2.2.2 Intersection

The intersection between two fuzzy sets(A and B) is described by the membership function $\mu_{A \cap B}(u)$ and is defined for all $u \in U$ by:

$$\mu_{A \cap B}(u) = \min(\mu_A(u), \mu_B(u))$$

2.3 Linguistic Variables

A linguistic variable is a variable that, instead of numerical values, consists of linguistic terms. Consider the linguistic variable *speed* which may consist of the terms *slow*, *medium* and *fast*. These linguistic terms can each be described using a fuzzy set. As an example we might consider the speed *slow* to be below 40 mph, *medium* to be around 55 mph and *fast* to be above 70 mph. The membership functions of these fuzzy sets can be seen in figure 1.

Using linguistic variables in this way makes it possible to describe vague and ambiguous concepts in a way that is understandable by machines. This in turns means that calculations and a variety of operations can be performed on these.

3. FUZZY SYSTEMS

The main idea behind a fuzzy system is to use the concept of linguistic variables to make decisions based on fuzzy rules and thereby get a better response compared to a system using crisp values. A basic generalized layout of a fuzzy system can be seen in figure 2.

The main components in the fuzzy system are *Fuzzification interface*, *knowledge base*, *decision making logic* and *defuzzification interface*.

3.1 Fuzzification Interface

The purpose of the fuzzification interface is threefold, read the input data, scale the data to fit the appropriate universe and fuzzify the input data to appropriate linguistic variables that can be handled as fuzzy sets.

Scaling the input data to map against the universe appropriate for the system can be done in a couple of ways. Either by doing a discretization of the universe and mapping the input data to the appropriate segment or by normalizing the universe and assigning each membership functions with an explicit function [8].

After the input data has been mapped to the appropriate universe the firing strength of the membership functions is determined. This means that each input data is determined to belong by a certain degree to each of the different fuzzy sets.

3.2 Knowledge Base

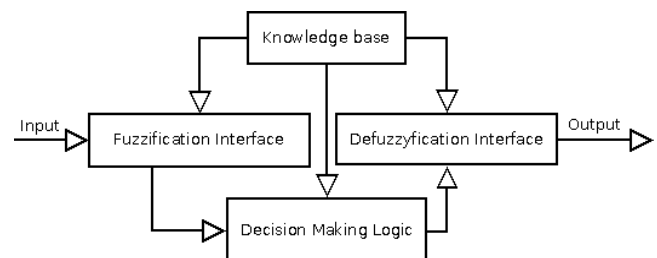
The knowledge base consists of a data base for the target area of the application and a linguistic control rule base. The data base contains expert knowledge in the form of definitions and other data that is necessary for the construction of the fuzzy system. This knowledge is often presented in the form of linguistic IF-THEN statements such as.

IF (a set of conditions are satisfied) THEN (infer a set of consequences)

These can, with the use of linguistic variables, be converted into fuzzy logical suggestions. Fuzzy rules can then be constructed from these statements.

3.3 Decision Making Logic

The decision making logic is the core of the fuzzy system. It is in this part of the system where the decision is made of how to interpret the output given from the fuzzy rules. Since several of the rules each give suggestions which may differ from each other it is important to have a system that is able to combine these suggestions into a decision of what the output should look like. How to combine the output from the fuzzy rules depends on how the system is to behave but in the case where the output from the fuzzy rules should have equal weight in determining the output of the system a logical AND connective can be used.



[8]Figure 2: A basic configuration of a fuzzy system

3.4 Defuzzification Interface

The purpose of the defuzzification interface is to translate the current fuzzy output into a reasonable control output. This can be done in several ways from simply choosing the set with the highest membership or using a variety of methods [9] such as Center of Gravity or Center of Area. In the case of a fuzzy classifier the defuzzification is where the fuzzy classification is turned into an actual classification.

4. GENETIC ALGORITHMS

The main idea behind GA is to simulate the process of biological evolution [10] and thereby getting a gradual improvement in the performance in regards to the problem. To do this a specific solution to the problem to be solved is treated as an individual in a population where the population is the collection of several solutions. The population is then allowed to evolve by choosing individuals that have the best solution to the problem and recombining them with each other. A new population is then created from a combination of the offspring and the old population. In this way better solutions will appear as the population “evolves”.

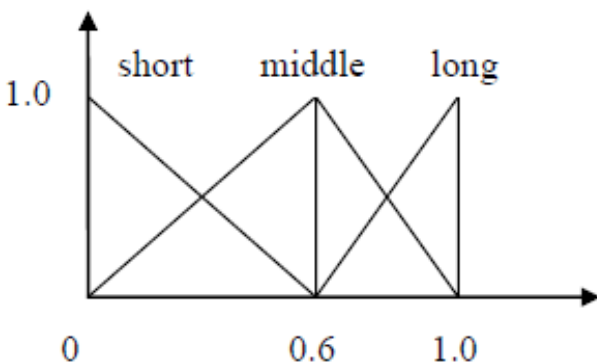
4.1 Fitness

To determine which individuals to choose from the original population there is a need to determine the fitness of each individual. The fitness of an individual is basically a measure of the individual’s ability to solve the problem. How this is done depends on what the problem that is to be solved looks like. A way of doing it is running the parameters from the solution through a series of test cases, looking at the outcome and determining how successful the individual is.

4.2 Mating

Based on the fitness of the individuals they are chosen to be part of a mating pool. The same individual can be part of the mating pool several times representing a particularly successful individual. However the individuals’ fitness does not guarantee that an individual gets chosen or not but rather, having a higher fitness value increases the chance of being selected to the mating pool.

Once a mating pool has been created the children of the old generation can be calculated. It is done by having pairs, from the mating pool, of individuals combine their solutions into two new



[15]Figure 3: The membership functions of the fuzzy sets short, middle and long

solutions. The chance that a pair of individuals will be recombined is determined by the *crossover rate*. How the individuals are combined depends on how the individuals are implemented to represent the individual solutions. In the case where the solution is represented by a string of binary or real values a simple exchange of some of the values is a possible way. A string consisting of real values may also be combined using a linear combination.

4.3 Mutation

To maintain genetic diversity in the population the newly created individuals has a chance of being mutated. This is just a slight change in each of the individuals to prevent the population from becoming all too similar.

4.4 Next Generation

The fitness value of the newly created individuals are then determined. These are then compared to the old population’s fitness values and the individuals with the highest fitness values of both the old and new populations are then chosen to create the next generation. The next generation is generally of equal size as that of the old one. The entire process is then restarted until a desired fitness has been reached or until a predetermined number of generations have passed.

5. IMPLEMENTATION

In this section the implementation of a fuzzy classifier will be presented. Starting off with the initial implementation of the fuzzy classifier and then continuing with the implementation of GA to improve the performance.

The iris data set consists of 150 samples 50 each specie, Setosa, Versicolor and Virginica. Each sample consists of four different input parameters, sepal length, sepal height, petal length and petal height as well as what species a particular sample belongs to.

From [15] the membership functions of the fuzzy sets to be used are known, these are shown in figure 3. A couple of fuzzy rules that are considered to be defined by experts in the area are to be used to classify the iris data [15].

$$\begin{aligned}
 R_1: & \text{IF } (x_1 = \text{short} \vee \text{long}) \text{ and } (x_2 = \text{middle} \vee \text{long}) \\
 & \text{and } (x_3 = \text{middle} \vee \text{long}) \text{ and } (x_4 = \text{middle}) \\
 & \text{THEN iris} = \text{versicolor} \\
 R_2: & \text{IF } (x_3 = \text{short} \vee \text{middle}) \text{ and } (x_4 = \text{short}) \\
 & \text{THEN iris} = \text{setosa} \\
 R_3: & \text{IF } (x_2 = \text{short} \vee \text{middle}) \text{ and } (x_3 = \text{long}) \\
 & \text{and } (x_4 = \text{long}) \text{ THEN iris} = \text{virginica} \\
 R_4: & \text{IF } (x_1 = \text{middle}) \text{ and } (x_2 = \text{short} \vee \text{middle}) \\
 & \text{and } (x_3 = \text{short}) \text{ and } (x_4 = \text{long}) \\
 & \text{THEN iris} = \text{versicolor}
 \end{aligned}$$

Where x_1 is the sepal length, x_2 is the sepal height, x_3 the petal length and x_4 petal height.

5.1 Fuzzyfication Interface

The data from iris set is read and normalized using the following formula:

$$x'_i = \frac{x_i - \min(i)}{\max(i) - \min(i)}$$

Where $\max(i)$ and $\min(i)$ denote maximum and minimum value of attribute x_i respectively. [15]

To determine to what fuzzy sets the data belong to the following formulas where extrapolated from the membership functions from the graph in figure 3.

$$\mu_{Short}(x) = \begin{cases} 0; & x < 0 \\ \left(1 - \frac{x}{0.6}\right); & 0 \leq x \leq 0.6 \\ 0; & x > 0.6 \end{cases}$$

$$\mu_{Middle}(x) = \begin{cases} 0; & x < 0 \\ \frac{x}{0.6}; & 0 \leq x \leq 0.6 \\ \left(1 + \frac{0.6}{0.4}\right) - \frac{x}{0.4}; & 0.6 < x \leq 1 \\ 0; & x > 1 \end{cases}$$

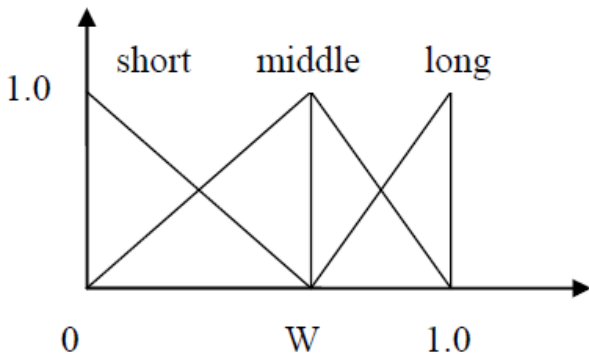
$$\mu_{Long}(x) = \begin{cases} 0; & x < 0.6 \\ \left(\frac{x}{0.4} - \left(\frac{1}{0.4} - 1\right)\right); & 0.6 \leq x \leq 1 \\ 0; & x > 1 \end{cases}$$

During the fuzzification process each input parameter will be mapped to the three fuzzy sets using these specific membership functions.

5.2 Knowledge Base

The underlying expert knowledge is contained within the fuzzy rules, r_1 , r_2 , r_3 and r_4 . Since the rules are constructed as a series of logical statements they are straightforward in their implementation. As an example in how this was achieved observe fuzzy rule number 2.

$$R_2: IF (x_3 = short \vee middle) \text{ and } (x_4 = short) \\ THEN iris = setosa$$



[16]Figure 4: The variable membership functions of short, middle and long

Since the logical or-function can be seen as a union between sets and a logical and-function can be seen as an intersection between sets, using the definitions of these fuzzy operators, the conditional part of the rule may be written as.

$$\text{Min}(\text{Max}(X_{3\text{short}}, X_{3\text{middle}}), X_{4\text{short}}) \quad (1)$$

Where $X_{3\text{short}}$, $X_{3\text{middle}}$ and $X_{4\text{short}}$ represents the degree of belonging by X_3 to the fuzzy set short, X_3 to the fuzzy set middle and X_4 to the fuzzy set short respectively.

The value received from this calculation will be in the interval $[0, 1]$ and represent the degree of belonging of the current set of input data to the specific flower specie to which the rule corresponds.

Each rule will result in an output that is a set that is as follows.

$$F_R = \left\{ \frac{a}{\text{Setosa}}, \frac{b}{\text{Versicolor}}, \frac{c}{\text{Virginica}} \right\}$$

Where a, b and c is the degree to which the rule determines that the current input parameters belong to the species Setosa, Versicolor and Virginica respectively. In the case of the example above with rule number 2, b and c would be zero while a would equal the resulting value of the calculations in (1).

5.3 Decision Making Logic

At worst there are four suggestions, one from each fuzzy rule, at which to choose from. Since each rule will give a set as output it is possible to combine these sets into a single set. This is done by applying a logical or-function to the outputted sets from the fuzzy rules as follows.

$$F = F_{R1} \vee F_{R2} \vee F_{R3} \vee F_{R4} = \left\{ \frac{A}{\text{Setosa}}, \frac{B}{\text{Versicolor}}, \frac{C}{\text{Virginica}} \right\}$$

The final set F will then contain the combined suggestions of the four fuzzy rules. From this, the current input parameters are considered to belong to the specie which has the highest degree of membership. This is done by comparing the values of A, B and C and choosing the corresponding specie based on which of the variables have the highest value.

5.4 GA Implementation

This subsection will focus on the how the GA implementation is done. To improve the fuzzy classifier the membership functions will be improved. To do this the model seen in figure 4 shows the new membership functions dependent on W. From these the following specific functions have been determined.

$$\mu_{Short}(x) = \begin{cases} 0; & x < 0 \\ \left(1 - \frac{x}{w}\right); & 0 \leq x \leq w \\ 0; & x > w \end{cases}$$

$$\mu_{Middle}(x) = \begin{cases} 0; & x < 0 \\ \frac{x}{w}; & 0 \leq x \leq w \\ \left(\left(1 + \frac{w}{1-w} \right) - \frac{x}{1-w} \right); & w < x \leq 1 \\ 0; & x > 1 \end{cases}$$

$$\mu_{Large}(x) = \begin{cases} 0; & x < w \\ \left(\frac{x}{1-w} \right) - \left(\left(\frac{1}{1-w} \right) - 1 \right); & w \leq x \leq 1 \\ 0; & x > 1 \end{cases}$$

As can be seen the membership functions of the four parameters are dependent on the variable W. Since each test set contains four parameters the individuals of the populations where chosen to be implemented using a string of four real values between zero and one representing the four values of W for the length and height of the sepal and petal. These are initially randomized to simulate a random population.

5.4.1 Fitness

To determine the fitness of each individual the solution of the individual is tested against the entire iris data set. The fitness of the individual is then determined by how accurate its solution is in classifying the correct species in the iris data set. This value will be between zero and one where zero means that none of the sets of parameters were accurately classified and one mean that all of them where.

5.4.2 Selecting individuals for the mating pool

To determine which individuals that will be allowed to mate the fitness is calculated for each individual in the population. This is done by simply running each individual solution against the entire iris data set and determining the accuracy. Once the fitness of each individual is determined the probability of being selected for the mating pool for each individual is calculated using.

$$P(I_n) = \frac{Fitness(I_n)}{\sum_{i=0}^m Fitness(I_i)}$$

Where $P(I_n)$ is the probability of selection for individual n with the population size m and is calculated by dividing the fitness of individual n with the sum of the fitness' of the whole population. Since the probability of selection for an individual is based on its fitness in proportion to the sum of the fitness of all individuals in the population, the sum of probabilities will be one.

Using this probability the individuals are then selected using a method commonly referred to as *fitness proportionate selection* or *roulette wheel selection* [10]. The way this works is by randomizing a number between 0 and 1. Then comparing the first individual's probability of selection and the randomized value, if the probability is greater than the randomized value then the individual is selected. If not then the probability of the next individual is added to the first one and compared again. This

```

1: SET randomValue to a randomized value between 0 and 1
2: SET index to 0
3: SET probabilitySum to the probability of
   the first individual in the population
4: WHILE randomValue > probabilitySum DO
5:   INCREASE index by 1
6:   ADD the probability of individual[index] to probabilitySum
7: END WHILE
8: SAVE individual[index] to the matingpool

```

Figure 5: Pseudo code for the roulette wheel selection algorithm

continues until the sum of probabilities is greater than the randomized value. An implementation of this is shown in pseudo code in figure 5.

This is done a number of times equal to the desired size of the mating pool. In this implementation the mating pool is the same size as the population.

5.4.3 Recombination

Once a mating pool has been created pairs of individuals are created. Individuals where paired together by individually pairing the first half of the mating pool with the inverted second half. The first individual is paired with the last individual, second first with the second last et cetera.

Once pairs have been chosen each pair of individuals may be recombined into a new pair. This is determined by the crossover rate. The crossover rate is the probability that a pair will be recombined. In the case where the pair isn't recombined the individuals are transferred as they are to the next stage.

In the case of a recombination, two new individuals are created by a linear combination of the two strings of real values representing the solution of the parents. The two new strings are created using the following formulas.

$$X_{NEW} = a_1 X_{OLD} + (1 - a_1) Y_{OLD}$$

$$Y_{NEW} = a_2 X_{OLD} + (1 - a_2) Y_{OLD}$$

Where X_{OLD} and Y_{OLD} are the parent strings, X_{NEW} and Y_{NEW} are the children strings and a_1 and a_2 are randomized values.

5.4.4 Mutation

When the new population has been created each individual in the population is mutated. The mutation is done by randomizing a value between minus one and one. This is then multiplied with a program parameter, mutation multiplier, to limit the effect of the mutation. This is then added to each value in an individual's string of real values to create a new mutated string. This calculation is done for each individual in the new population.

5.4.5 Next generation

Creating the next generation is done by comparing the fitness individuals of the old population with that of the fitness of the newly recombined and mutated population. Out of these two populations the individuals with the best fitness are selected one by one until a new population equal in size to the original population has been created.

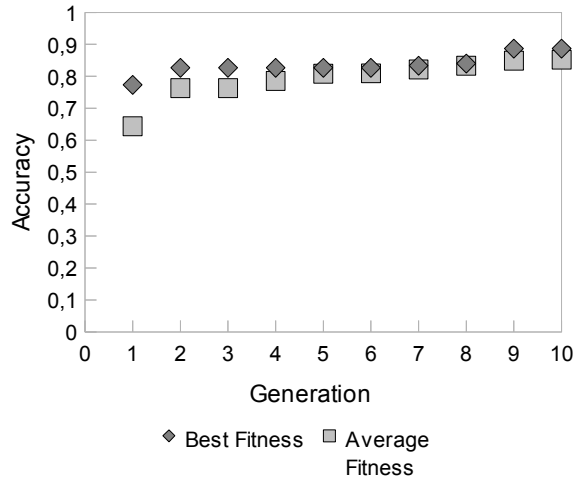


Figure 6: The accuracy of the fuzzy classifier as generations progress with a population of 5

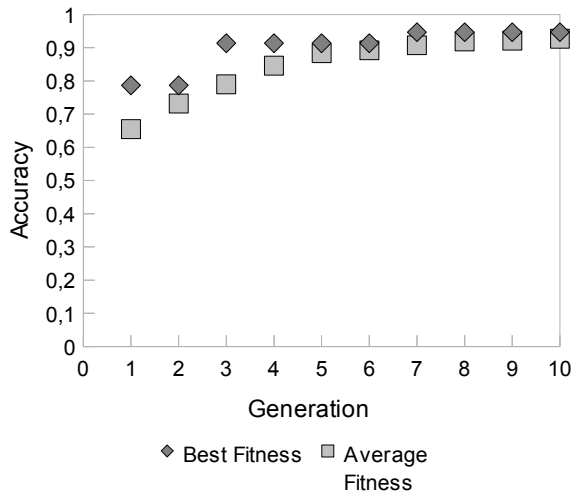


Figure 7: The accuracy of the fuzzy classifier as generations progress with a population of 10

6. RESULTS

The initial implementation of the fuzzy classifier, without the improvement from GA, manages to accurately classify 115 of the 150 data sets which equals roughly to an accuracy of around 77%.

With the improvement from GA the accuracy of the classifier is significantly improved. Up to 143 out of the 150, circa 95%, sets can be accurately classified after the algorithm has been allowed to run its course. Since it is possible to vary certain program parameters, below are three graphs showing the progress of the GA during the first ten generation using a population size of 5, 10 and 20. In all of these runs a crossover rate of 90% was used and a mutation multiplier of 0.05. The GA was allowed to run 10 generations. In the cases where the population size is 10 or 20 it was determined that Further generations did not improve the results. The case with a population size of five is given as a comparison.

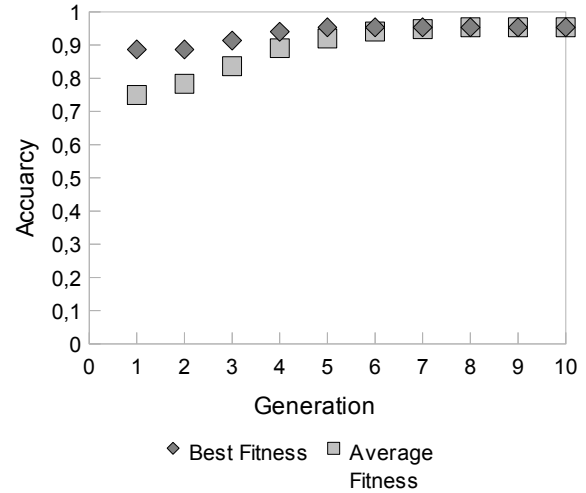


Figure 8: The accuracy of the fuzzy classifier as generations progress with a population of 20

7. CONCLUSIONS

As can be seen in figures 6, 7, 8 after a few generations the increase in accuracy of the fuzzy classifier is quite noticeable, from 77% to 95%, compared to the initial implementation. This that when you have access to reliable expert data it is possible to improve fuzzy classifiers using genetic algorithms. Comparing the results to other fuzzy classifiers, seen in Table 1, there is room for further improvement.

Table 1: Comparison of results

Classifier	Accuracy
This paper	95,30%
X. Chang et al. [2]	99,30%
H. Ishibuchi et al. [6]	94,67%
C.Rani et al. [12]	100,00%

To improve the accuracy further a couple of approaches exist. The most straightforward one would be to look at the fuzzy rules and possibly make changes to them in order to improve the accuracy further. This could possibly be an area where future work could take place.

What is interesting to note on the three different population sizes is that as the initial best fitness of the population is increased as the population size increases. This is probably because of the limited size of the universe which means that with a limited set of possible solutions it is possible to get a good solution with the initial randomization. With a larger population a there is a larger chance of getting at least one good solution.

8. ACKNOWLEDGMENTS

I would like to thank Ralf Strömberg, student at Mälardalen University, together with whom I did the work that is presented in this paper.

9. REFERENCES

- [1] J. C. Bezdek, J. M. Keller, R. Krishnapuram, L. I. Kunecheva, N. R. Pal. 1999. *Will the Real Iris Data Please Stand Up?*. IEEE Transactions on Fuzzy Systems, volume 7, no. 3 June.
- [2] X. Chang, J.H. Lilly. 2004. *Evolutionary Design of a Fuzzy Classifier from Data*. IEEE Transactions on Systems, Man and Cybernetics, Part B. Volume 34. Issue 4. pp. 1894-1906
- [3] B. Carse, T. C. Fogarty, A. Munro. 1996. *Evolving fuzzy rule based controllers using Genetic Algorithms*. Fuzzy Sets and Systems, volume 80, issue 3, pp. 273-293.
- [4] O. Cordón; F. Gomide; F. Herrera; F. Hoffman; L. Magdalena. *Ten years of Genetic Fuzzy Systems: Current Framework and New Trends*. Fuzzy Sets and Systems, volume 141, issue 1, pp. 5-31.
- [5] F. Herrera, M. Lozano, J. L. Verdegay. 1995. *Tuning fuzzy logic controllers by genetic algorithms*. International Journal of Approximate Reasoning, volume 12, issues 3-4, April-May, pp. 299-315.
- [6] H. Ishibuchi, K. Nozaki, N. Yamamoto, H. Tanaka. 1995. *Selecting fuzzy if-then rules for classification problems using genetic algorithms*. IEEE Transactions on Fuzzy Systems, volume 3, issue 3, pp. 260-270.
- [7] L.I. Larkin. 1984. *A fuzzy logic controller for aircraft control*. The 23rd IEEE Conference on Decision and Control, pp. 894-897.
- [8] C. C. Lee. 1990. *Fuzzy Logic in Control Systems: Fuzzy Logic Controller -Part 1*. IEEE Transactions on Systems, Man. And Cybernetics, volume 20, no. 2 March/April.
- [9] Van Leekwijck and Kerre. 1999. *Defuzzification: criteria and classification*. Fuzzy Sets and Systems, volume 108, pp. 159-178.
- [10] Tom M. Mitchell. 1997. *Machine Learning*. Singapore: McGraw-Hill
- [11] K. Oishi, M. Tominaga, A. Kawato, Y. Abe, S. ImayasuA. Nanba. 1991. *Application of fuzzy control theory to the sake brewing process*. Journal of fermentation and Bioengineering, volume 72, issue 2, pp. 115-121.
- [12] C.Rani, S.N.Deepa. 2010. *Designing of Optimal Fuzzy Classifier System Using Particle Swarm Optimization*. International Conference on Innovative Computing Technologies. pp.1-6
- [13] L. A. Zadeh. 1965. *Fuzzy sets*. Information and Control, vol 8, pp. 338-353.
- [14] Y-S. Zhou, L-Y. Lai. 1995. *Optimal Design for fuzzy controllers by genetic algorithms*. International Conference on Industrial Automation and Control: Emerging Technologies. Pp. 429-435.
- [15] *Assignment 2*. 2008, Task description, School of Innovation, Design and Engineering, Mälardalen University, viewed on

23 of November 2010,

<http://www.idt.mdh.se/kurser/cdt407/nxg01/2008p1/labs/la_b2.pdf>

- [16] *Assignment 3*. 2008, Task description, School of Innovation, Design and Engineering, Mälardalen University, viewed on 23 of November 2010, <http://www.idt.mdh.se/kurser/cdt407/nxg01/2008p1/labs/la_b3.pdf>

10. APPENDIX

This section of the paper contains the data for the graphs used in the paper.

10.1 Data for figure 6

Generation	Average Fitness	Best Fitness
1	0,644	0,773333
2	0,762667	0,826667
3	0,762667	0,826667
4	0,785333	0,826667
5	0,808	0,826667
6	0,809333	0,826667
7	0,821333	0,833333
8	0,833333	0,84
9	0,849333	0,886667
10	0,852	0,886667

10.2 Data for figure 7

Generation	Average Fitness	Best Fitness
1	0,654667	0,786667
2	0,732	0,786667
3	0,789333	0,913333

4	0,846	0,913333
5	0,88333	0,913333
6	0,892	0,913333
7	0,907333	0,946667
8	0,918667	0,946667
9	0,921333	0,946667
10	0,927333	0,946667

10.3 Data for figure 8

Generation	Average Fitness	Best Fitness
1	0,749333	0,886667
2	0,783	0,886667
3	0,83633	0,913333
4	0,890333	0,94
5	0,92	0,953333
6	0,940666	0,953333
7	0,947667	0,953333
8	0,953333	0,953333
9	0,953333	0,953333
10	0,953333	0,953333