International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 07 | July - 2022 Impact Factor: 7.185 ISSN: 2582-3930

Grading System Using Fuzzy Logic

Mohammad Faraz Deshmukh

Master of Computer Applications,

Vivekanand Education Society's Institute of Technology,

University of Mumbai

ABSTRACT

The collaboration between e-learning activities and activities that provide formative and comprehensive assessments are receiving a lot of attention in the educational world. These kind of activities can be implemented for online evaluation of candidates. There is no doubt that all institutions rather be large or small, use some kind of protocols in their field for assessing the performance of traditional regular and e-learning students. This paper proposes an efficient, automated and intelligent approach to calculate or assees the overall performance of learning candidates using Fuzzy Logic System. This system takes parameters such as attendance, internal marks, programming lab practicals, and total assessment include mid and final grades. The result shows that the Expert System for assessing the overall performance of candidates evaluates efficiently. The proposed model also affects the educational institution's rating system and improvement of educational background by formative and comprehensive student evaluation analysis.

INTRODUCTION

The student's education success or level is considered based on their achievements of grades in exams and other academic activities University. The IT industry takes this into account and rapidly progress. As there is a big gap in the IT industry and educational institutions, therefore, the need for automation, intelligent and efficient system for calculation or evaluation of computer science student performance or a computer programming course. Formative assessment (FA) is an assessment used in classes to get information of progress of student. Using this information, the teacher master their teaching methods to improve learning and students are aware of them learning process. You can use online tools implement these learning evaluation activities or formative assessment in laboratory exercises. This type of evaluation is not same as intermediate assessment held several times during class to evaluate both the educational program and the student's ongoing performance and from Summative Assessment (SA) that is carried out at the end of the curriculum to recognize learning of candidates.

This paper is all about the development process of such expert system that has the only purpose of it which conveys methods and technology used for development which evaluates the overall performance of all candidates courses at educational institutions and universities.

LITERATURE REVIEW

Systematic review was held for learning style that matches e-learning students problem. In front of us, changes to e-learning content there are multiple e-learning errors. The author discusses and review the effectiveness of learning styles and different classification methods in different e-learning courses Elsevier, Springer, Wiley, PubMed resource. Other reviewed work referenced Feedback is provided to e-learning students system. The author got papers from 2008 to 2019 based on IEEE Explore Digital Library which helped to focus on personalized feedback. The author also discussed challenges regarding work-in progress through fuzzy sets that were explored as a measurement.

In educational institutes, the conventional method is being used on a large scale to calculate or assess the performance of students of a specific Computer Science (CS), Computer Programming (CP) or Software Engineering (SE) courses. In the conventional method, a teacher or instructor takes the final results of all academic activities of a student of a specific course and puts time in analyzing those results to come up with the linguistic value of the performance of that student in that specific course. If there are few students of one discipline that might be easy but if there are hundreds or thousands of students of more than one discipline that is almost impossible for one teacher or instructor to handle it. The author claims that it will require a lot of human effort and human resource to handle all of it manually and thus an automatic, intelligent and efficient system is required that can easily handle this situation The author focused on computer science with computer programming and proposed that scholarship could be improved description performance by modeling. The author focused on more effective fuzzy-based approach system, introducing an intelligent e-learning system based on learning environment and fuzzy logic showing student knowledge. It also developed a system using fuzzy logic to evaluate the performance of the calculated subjects of VESIT students with the help of MATLAB. They have been more focused on the value of performance than what the usual methodology focuses on system.

Take two inputs. With subject A and subject B generates the output as a power value. The researched and developed system was used for evaluation performance of 22 students, the system was successful. Another proposed system, evaluate student activity in the laboratory engineering education application. They suggested a new approach to student grades, “Improvement of the process using peer assessment, group and individual evaluation methods.” Fuzzy helps you make more reliable decisions, evaluation of the application for a given list standard. This research shows that it is ambiguous and the system can provide a better rating system than classic or traditional system The author too discussed the overall results of the computer science courses with fuzzy logic approach for assessment and evaluation. Another paper discussed learning modality for students and interaction activities, especially for learming computer programming by the implementation of fuzzy logic. The author suggested more combinations of fuzzy theory and cognitive theory effective for adaptive e-assessment. The author also introduced a hybrid approach with convolution of Neural network (CNN) and fuzzy system for face recognition representation in an adaptive e-learning environment. It also monitored the learning performance of each and every candidate.

FUZZY LOGIC : ALGORITHM

One of the classification methods is to use Fuzzy Logic Technique to classify the subject. The nature of Fuzzy Logic is different from other methods. It is used when the answer to the classification question is more versatile than just having 'Yes' or 'No' or similar two variations. For example, 'How are you?' usually has two variations which are 'Good' and 'Not Good', but if the requirement is to keep it more versatile and to have more than two variations such as 'Very Bad', 'Bad', 'I don't know', 'Good', 'Very Good', then it becomes an application of Fuzzy Logic. In Mathematics and Computer Science, the researchers have designed a special algorithm that follows the workflow to implement fuzzification and defuzzification, which are the core aspects of Fuzzy Logic Methodology.

- **Defining Input and Output Variables (Crisp Values)**

The very first step in the FL Algorithm is to define the input variables that are going to be passed to the algorithm as input parameters.
Example input variables are shown in Table 1.

Table 1. Example of Input Variables	
Input Variables	**Description**
p	Input variable one
q	Input variable two
r	Input variable three

Next step is to define the output variable. Table 2 shows the example output variable.

Table 2: Example of Output Variable	
Output Variable	**Description**
z	Output variable

Next step is to assign the numeric range (crisp values) to input and output variables. Table 3 shows the example range assigned to the defined variables.

Table 3. Example of Assigned Range to Defined Variables		
Variables	**Min Range Value**	**Max Range Value**
P	1	10
Q	1	15
R	1	20
Z	1	10

- **Fuzzification**

Next step is to assign the linguistic words to the variables at the specific span of the range. Later, the triangular membership function is applied to each variable to get Fuzzification of the target variables [22]. Mathematically, triangular membership function is defined as below:

$$f(x,a,b,c) = \begin{cases} 0, & x \le a \\ \frac{x-a}{b-a}, & a \le x \le b \\ \frac{c-x}{c-b}, & b \le x \le c \\ 0, & c \le x \end{cases}$$

It can also be defined in the other way as:

$$f(x,a,b,c) = max\left(min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$$

Code Snippet I, II, III shows the Pseudo Code of assigning example linguistic words to input variables ‘p’, ‘q’ and ‘r’ respectively.

Code Snippet I: Example Pseudo Code of Assigning Linguistic Words to Input Variable One

```
IF p is greater or equal to 1 and p is smaller
or equal to
3.3
THEN Set p to the string ‘BAD’
ELSE IF p is greater than 3.3 and p is smaller
than 6.6
THEN Set p to the string
‘AVERAGE’
ELSE
Set p to the string
‘GOOD’
END IF
```

Code Snippet II: Example Pseudo Code of Assigning Linguistic Words to Input Variable Two

```
IF q is greater or equal to 1 and q is smaller
or equal to 5
THEN Set q to the string
‘BAD’
ELSE IF q is greater than 5 and q is smaller
than 10
THEN Set q to the string
‘AVERAGE’
ELSE
Set q to the string ‘GOOD’
END IF
```

Code Snippet III: Example Pseudo Code of Assigning Linguistic Words to Input Variable Three

```
IF r is greater or equal to 1 and r is smaller
or equal to
6.7
THEN Set r to the string ‘BAD’
ELSE IF r is greater than 6.7 and r is
smaller than 13.4
THEN Set r to the string
‘AVERAGE’
```

```
ELSE
Set r to the string 'GOOD'
END IF
```

Now to fuzzify each variable, triangular membership function is applied.

- **Defuzzification**

The same triangular function is used for output variable 'z' as defuzzification [23]. Code Snippet IV shows the Pseudo Code of assigning example linguistic words to variable 'z'.

Code Snippet IV: Example Pseudo Code of Assigning Linguistic Words to Output Variable

```
IF z is greater or equal to 1 and z is smaller
or equal to
6.7
THEN Set z to the string 'BAD'
ELSE IF z is greater than 6.7 and z is
smaller than 13.4
THEN Set z to the string
'AVERAGE'

ELSE
Set z to the string 'GOOD'
END IF
```

- **Defining the Fuzzy Inference Rules**

Fuzzy Inference Rules (FLS) calculates the output based on the rules known as Fuzzy Inference Rules. These rules are based on IF-THEN statements and define the output based on different variations of inputs [24]. The following formula is used to know the number of rules:

$r = m^n$

In the above formula, 'r' is the total number of rules, 'm' is the total input linguistic words, and 'n' is the total input variables. As per the current example, the number of rules is:

$r = 3^3 = 27$

This means, a total of 27 unique rules should be defined and the order does not matter thus rules can be defined in any order. The things to be noted here are, each rule must be unique and must not repeat. Table 4 shows the Fuzzy Inference Rules for the current example.

Table 4. Example of Fuzzy Inference Rules

#	p	q	R	Z
1	BAD	BAD	BAD	BAD
2	BAD	BAD	AVERAGE	BAD
3	BAD	BAD	GOOD	BAD
4	BAD	AVERAGE	BAD	BAD
5	BAD	AVERAGE	AVERAGE	AVERAGE
6	BAD	AVERAGE	GOOD	AVERAGE
7	BAD	GOOD	BAD	BAD
8	BAD	GOOD	AVERAGE	AVERAGE
9	BAD	GOOD	GOOD	AVERAGE
10	AVERAGE	BAD	BAD	BAD
11	AVERAGE	BAD	AVERAGE	AVERAGE
12	AVERAGE	BAD	GOOD	AVERAGE
13	AVERAGE	AVERAGE	BAD	AVERAGE
14	AVERAGE	AVERAGE	AVERAGE	AVERAGE
15	AVERAGE	AVERAGE	GOOD	AVERAGE
16	AVERAGE	GOOD	BAD	AVERAGE
17	AVERAGE	GOOD	AVERAGE	AVERAGE
18	AVERAGE	GOOD	GOOD	GOOD
19	GOOD	BAD	BAD	BAD
20	GOOD	BAD	AVERAGE	AVERAGE
21	GOOD	BAD	GOOD	AVERAGE
22	GOOD	AVERAGE	BAD	AVERAGE
23	GOOD	AVERAGE	AVERAGE	AVERAGE
24	GOOD	AVERAGE	GOOD	GOOD
25	GOOD	GOOD	BAD	AVERAGE
26	GOOD	GOOD	AVERAGE	GOOD
27	GOOD	GOOD	GOOD	GOOD

These rules were generated using a program written in Python 3.8. Code Snippet V shows the Python code that generated the Fuzzy Inference Rules. The code can be customized according to the required generating rules.

Code Snippet V:Program Written in Python to Generate Fuzzy Rules

```
1.      ling_words   =   ("bad",   "average",
    "good")
2.    def rule(i, *args):
3.
4.       a = ""
5.
6.           for i in range(len(args)):
7.               b = ling_words[args[i]] + '
    '
8.           a = a + b
9.
10.      if a.count("good") == 3
    or (a.count("good") == 2 and
    a.count("average") == 1):
11.          return a + 'good'
12.         elif  a.count("bad")  ==  3  or
    (a.count("bad")      ==      2      and
    a.count("average")     ==     1)     or
    (a.count("bad") == 2 and
    a.count("good") == 1):
13.          return a + 'bad'
14.      elif a.count("average")
    >= 3 or (a.count("average") ==
    2 and a.count("good") == 1) or
    (a.count("average") == 2 and
    a.count("bad") == 1) or (a.count("good")
    == 2 and
    a.count("bad") == 1):
15.          return a + 'average'
16.      else:
17.          return a + 'average'
18.   i = 0
19.
20.  for p in range(0, 3):
21.      for q in range(0, 3):
22.          for r in range(0, 3):
23.                  print (i + 1, rule(i, p,
    q,
```

Next, these Fuzzy Rules are passed to the system as an Inference Engine, which is then used to generate and defuzzify the output.

RESEARCH METHODOLOGY

- **Assigning Linguistic Words to the Defined**

Input Variables (Fuzzification)

Three linguistic words were decided to be assigned to input variables viz. 'Bad', 'Average', 'Good'. Triangular membership function was used to fuzzify the inputs. Code Snippet VII shows the Python code that was used to assign linguistic words to input variables and fuzzify them using triangular membership function.

Code Snippet VII: Program Written in Python to Assign Linguistic Words to Input Variables and Fuzzify Them

```
1.       p['bad'] =
fuzzy.trimf(p.universe, [0,
 4.5, 8])

                        p['average']    =
fuzzy.trimf(p.universe, [7,
11.5, 15])

3.       p['good'] =
fuzzy.trimf(p.universe, [14,
17.5, 20])
4.
q['bad'] =
fuzzy.trimf(q.universe, [0,
6.5, 13])
q['average'] = fuzzy.trimf(q.universe,
[12,
19.5, 26])
q['good'] =
fuzzy.trimf(q.universe, [25,
32.5, 40])
8.
r['bad'] =
fuzzy.trimf(r.universe, [0,
4.5, 8])
r['average'] = fuzzy.trimf(r.universe,
[7,
12.5, 16])
```

```
r['good']   =   fuzzy.trimf(r.universe,
[15,
20.5, 25])
12.
s['bad'] = fuzzy.trimf(s.universe, [0,
7.5, 15])
s['average'] = fuzzy.trimf(s.universe,
[14,
22.5, 30])
s['good']   =   fuzzy.trimf(s.universe,
[29,
37.5, 45])
16.
t['bad'] = fuzzy.trimf(t.universe, [0,
15,
30])
t['average'] = fuzzy.trimf(t.universe,
[29,
45, 60])
t['good']   =   fuzzy.trimf(t.universe,
[59,
75, 90])
```

The 'fuzzy' in the above code is an alias of 'skfuzzy' (Python's scikit-fuzzy library name), and trimf() method is an implementation of the triangular membership function. There are other membership functions as well such as, Trapezoidal Membership function and Sigmoid Membership function, but for this work Triangular Membership function is used.

- **Assigning Linguistic Words to the Defined Output Variable (Defuzzification)**

Five linguistic words were decided to be assigned to output variable viz. 'Poor', 'Bad', 'Average', 'Good', 'Excellent'. Triangular membership function was also used for the output variable. Keep in view the fact that we have to use the same membership function for both input variables and the output variable. Code Snippet VIII shows the Python code that was used to assign linguistic words to input variables and fuzzify them using triangular membership function.

Code Snippet VIII: Program Written in Python to Assign Linguistic Words to Output Variable and Defuzzify Them

```
z['poor'] =
fuzzy.trimf(z.universe, [0, 1,
2])
z['bad'] =
fuzzy.trimf(z.universe, [1,
2.5, 4])
z['average'] = fuzzy.trimf(z.universe, [3,
4.5, 6])
z['good'] =
fuzzy.trimf(z.universe, [5,
6.5, 8])
z['excellent']   =   fuzzy.trimf(z.universe,
[7,
8.5, 10])
```

- **Defining Fuzzy Inference Rules**

Fuzzy Logic System calculates the output based on rules which should be defined by the developer and these rules are known as Fuzzy Inference Rules. Total numbers of Fuzzy Rules are given as:

$$r = m^n$$
$$r = 3^5 = 243$$

Table 9 shows the first ten fuzzy rules of FLS AI Algorithm.

Table 9. First Ten Fuzzy Inference Rules from FLS AI Algorithm

#	p	Q	r	s	t	z
1	Bad	Bad	Bad	Bad	Bad	Poor
2	Bad	Bad	Bad	Bad	Average	Poor
3	Bad	Bad	Bad	Bad	Good	Bad
4	Bad	Bad	Bad	Average	Bad	Poor
5	Bad	Bad	Bad	Average	Average	Bad
6	Bad	Bad	Bad	Average	Good	Bad
7	Bad	Bad	Bad	Good	Bad	Bad
8	Bad	Bad	Bad	Good	Average	Bad
9	Bad	Bad	Bad	Good	Good	Bad
10	Bad	Bad	Average	Bad	Bad	Poor

All of the 243 Fuzzy Rules were stored in a Knowledge Base text file whose extensions are '.kb'. The rules were then passed to an Inference Engine, which produced an output based on these rules. Code

Snippet IX shows the Python code that was used to pass all of the Fuzzy Rules in Inference Engine and generate an output.

Code Snippet IX: Program Written in Python to Process the Fuzzy Rules as an Inference Engine

```
rls = []

with open(RULES_abs_path,
"r") as fh:
    for line in fh:
        if line != '\n' and line != None:

            temp = line.split(" ")

            result                          =
control.Rule(p[temp[0]] & q[temp[1]] &
r[temp[2]] & s[temp[3]] & t[temp[4]],
z[temp[5].rstrip()])

            rls.append(result)

  perf_calc                     =
control.ControlSystem(rls)
```

CONCLUSION

The experimental studies conducted by Grading System using Fuzzy Logic evaluates performance very effectively. It can be inferred that as a substitute for degenerate human resources time to evaluate the performance of thousands of people if students do it manually, we recommend using Grading System using Fuzzy Logic software that evaluates performance of candidates on a large-scale. It is also beneficial to save the time and energy of teachers and professors and can enhance the pedagogy in the future for candidate performance enhancement.

REFERENCES

L. M. De Campos and S. Moral, "Learning rules for a fuzzy inference model," Fuzzy Sets and Systems.

D. Sinha and E. R. Dougherty, "Fuzzification of set inclusion: theory and applications," Fuzzy sets and systems.

E. Panadero, H. Andrade, and S. Brookhart, "Fusing self-regulated learning and formative assessment: A roadmap of where we are, how we got here, and where we are going," The Australian Educational Researcher.

W. Van Leekwijck and E. E. Kerre, "Defuzzification: criteria and classification," Fuzzy sets and systems.

K. Chrysafiadi, C. Troussas, and M. Virvou, "Combination of fuzzy and cognitive theories for adaptive e-assessment," Expert Systems with Applications.

J. W. Gikandi, D. Morrow, and N. E. Davis, "Online formative assessment in higher education: A review of the literature,"

A. Khamparia and B. Pandey, "Association of learning styles with different e-learning problems: a systematic review and classification," Education and Information Technologies.

W. Harlen and M. James, "Assessment and learning: differences and relationships between formative and summative assessment," Assessment in Education: Principles, Policy & Practice.