A Case Study in Using Fuzzy Logic to Combine Student's Opinion of Teaching (SOT) into a Final Defuzzified Result

Kuo-Pao Yang, Patrick McDowell, Charles Norsworthy, Damodar Dahal, Andrew Burningham, Matthew Boudreaux
Computer Science Department
Southeastern Louisiana University
Hammond, LA 70402 USA

***Abstract*— This paper discusses a possible way to combine student's inputs on a Student's Opinion of Teaching (SOT) into an overall result using fuzzy logic to create a defuzzified result that simulates a curbed score on a test.**

Keywords*— *fuzzy logic; fuzzy logic system; student opinion of teaching; defuzzified result

I. Introduction

Every semester, students fill out a questionnaire whose purpose is to find the student's opinion of teaching, or SOT. The SOT form is composed of several questions regarding the course, the instructor, the class environment, etc. This paper asks the question – "Is there a way to combine their inputs into a final, overall opinion of teaching?" The paper discusses a possible way to combine student's inputs on a student's SOT into an overall result using fuzzy logic to create a defuzzified result that simulates a curbed score on a test.

Fuzzy logic is an approach to computing based on "degrees of truth" rather than the traditional "true or false" (1 or 0) Boolean logic. During the recent years, fuzzy control has emerged as one of the most active and fruitful areas for research in the applications of fuzzy set theory, especially in the realm of industrial processes, which do not lend themselves to be controlled by conventional methods because of a lack of quantitative data regarding the input-output relations [1]. Fuzzy logic has demonstrated to be effective in many different practical applications, from linguistics [2], storage [3], nuclear reactor controls [4], and to even in legal interpretation [5]. Since students' opinions of teaching are subjective experiences, it is difficult to accumulate them into a single scalar value that accurately reflects the total of their opinions. This project explores the use of fuzzy logic in order to measure the students' opinions and combine them into a more truthful value, or figure of merit, that accurately reflects a teacher's rating.

This project provides an alternative method to answer the classic question of how to rate a course based on aspects such as professor, course topics, and classroom. This fuzzy method is compared to the traditional statistical based methods, which typically rely on averages or weighted averages. In short, this project takes a new approach to finding out what matters to students regarding learning in the classroom by employing fuzzy logic rather than traditional methods.

In the following sections, we review the related work on using fuzzy logic in evaluations inside universities. This paper describes the approach and implementation used to combine students' SOT into a final defuzzified result. Finally, this paper shows the results and provides ideas for future areas of research.

II. Background

This project is about using fuzzy logic to evaluate how students feel about courses taught at a university. There are previous attempts at using fuzzy logic in various evaluations. For example, an attempt has been made to evaluate students' performances with fuzzy logic based on three characteristics with three different exams [6]. Similarly, students' performances are also evaluated with three parameters using fuzzy logic [7]. Furthermore, five parameters are fuzzified into a fuzzy inference system for the student performance evaluation method [8][9]. Finally, fuzzy logic is used to analyze client satisfaction surveys due to the inherently imprecise nature of data from these surveys [10][11].

These examples of using fuzzy logic in evaluations do not address our basic dilemma of combining student answers on a SOT into a final, overall, score or figure of merit. There are no papers that address this specific problem. Also, the current SOT's here at Southeastern Louisiana University do not have building ratings and course ratings. Furthermore, using fuzzy logic with fuzzy rules and sets can allow for a more intuitive approach to creating an overall "curbed" score better than a simple weighted average of results. Fuzzy logic itself is often used with imprecise things, such as surveys asking questions like how friendly an instructor is.

III. Implementation

This project implements the solution using the programming language Python. It uses a fuzzy logic library in Python named SciKit Fuzzy. For each questionnaire, this system has the same calculations being used to compute the rounded score, with the last question being weighed less heavily than the other four variables.

There are five antecedents, or inputs, into the fuzzy logic system. Since the purpose of the program is to reuse it for multiple surveys, this system simply names these variables “antecedent1”, “antecedent2”, “antecedent3”, “antecedent4”, and “antecedent5”. The consequent to the system is one variable named “overall_rating.” Every input antecedent has the same range on the X-axis: 0 to 10. The consequent has a range from 2 to 14 on the X-axis. This is implemented in Python as shown in Fig. 1.

```
# define intervals for the antecedents and consequent
antecedent1 = ctrl.Antecedent(np.arange(0, 11, 1), 'antecedent1')
antecedent3 = ctrl.Antecedent(np.arange(0, 11, 1), 'antecedent3')
antecedent2 = ctrl.Antecedent(np.arange(0, 11, 1), 'antecedent2')
antecedent4 = ctrl.Antecedent(np.arange(0, 11, 1), 'antecedent4')
antecedent5 = ctrl.Antecedent(np.arange(0, 11, 1), 'antecedent5')
overall_rating = ctrl.Consequent(np.arange(2, 15, 1), 'overall_rating')
```

Fig. 1. *Python implementation of antecedent and consequent intervals*

Each input antecedent is composed of the same three fuzzy sets: “poor”, “average”, and “good” shown in Fig. 2. The consequent is composed of the fuzzy sets called “low”, “medium”, and “high” shown in Fig. 3. Each variable has the same membership functions for each fuzzy set, a triangle curve with a straight line running between coordinates (0, 0), (0, 1), and (5, 0) for “poor”, (0, 0), (5, 1), and (10, 0) for “average”, and (5, 0), (10, 1), and (10, 0) for “good.” The membership functions for the output variable are a trapezoid composed of straight lines running between coordinates (2, 0), (2, 1), (6, 1), and (7, 0) for “low”, a triangle composes of straight lines running between coordinates (2, 0), (7, 1), and (12, 0) for “medium,” and a trapezoid composes of straight lines between coordinates (7, 0), (8, 1), (12, 1), and (13, 0) for “high.” This is implemented in Python as shown in Fig. 4.

To visualize this, consider the following pictures shown in Fig. 2. This is a picture of the first antecedent, but every antecedent has membership functions identical to each other.

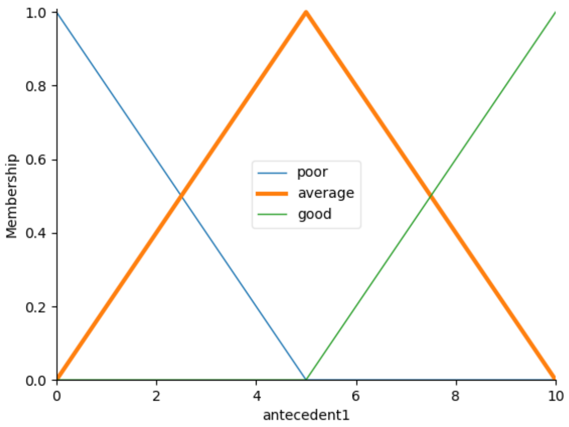


Fig. 2. *Graph of antecedent membership functions*

Next, consider the graph of the overall_rating consequent shown in Fig. 3.

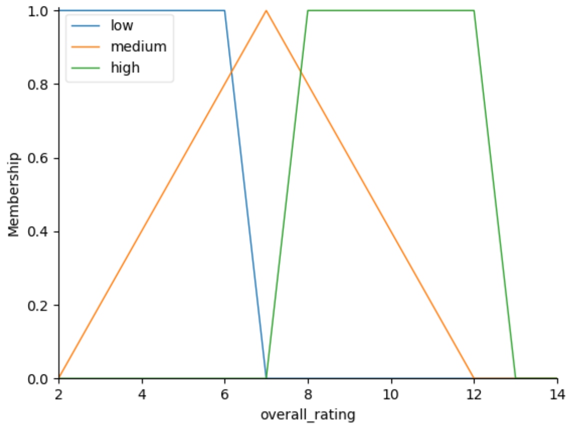


Fig. 3. *Graph of consequent membership function*

```
# define membership functions for antecedents and consequent
antecedent1['poor'] = fuzz.trimf(antecedent1.universe, [0, 0, 5])
antecedent1['average'] = fuzz.trimf(antecedent1.universe, [0, 5, 10])
antecedent1['good'] = fuzz.trimf(antecedent1.universe, [5, 10, 10])

antecedent2['poor'] = fuzz.trimf(antecedent2.universe, [0, 0, 5])
antecedent2['average'] = fuzz.trimf(antecedent2.universe, [0, 5, 10])
antecedent2['good'] = fuzz.trimf(antecedent2.universe, [5, 10, 10])

antecedent3['poor'] = fuzz.trimf(antecedent3.universe, [0, 0, 5])
antecedent3['average'] = fuzz.trimf(antecedent3.universe, [0, 5, 10])
antecedent3['good'] = fuzz.trimf(antecedent3.universe, [5, 10, 10])

antecedent4['poor'] = fuzz.trimf(antecedent4.universe, [0, 0, 5])
antecedent4['average'] = fuzz.trimf(antecedent4.universe, [0, 5, 10])
antecedent4['good'] = fuzz.trimf(antecedent4.universe, [5, 10, 10])

antecedent5['poor'] = fuzz.trimf(antecedent5.universe, [0, 0, 5])
antecedent5['average'] = fuzz.trimf(antecedent5.universe, [0, 5, 10])
antecedent5['good'] = fuzz.trimf(antecedent5.universe, [5, 10, 10])

overall_rating['low'] = fuzz.trapmf(overall_rating.universe, [2, 2, 6, 7])
overall_rating['medium'] = fuzz.trimf(overall_rating.universe, [2, 7, 12])
overall_rating['high'] = fuzz.trapmf(overall_rating.universe, [7, 8, 12, 12])
```

Fig. 4. *Python implementation of membership functions.*

Essential to any fuzzy logic system is its fuzzy rules. The fuzzy rules in this system are thus:

1. If antecedent1 is “poor”, then overall_rating is “low.”
2. If antecedent2 is “poor”, then overall_rating is “low.”
3. If antecedent3 is “poor”, then overall_rating is “low.”
4. If antecedent4 is “poor”, then overall_rating is “low.”
5. If antecedent5 is “poor”, then overall_rating is “low” with .50.
6. If antecedent1 is “average”, then overall_rating is “medium.”
7. If antecedent2 is “average”, then overall_rating is “medium.”
8. If antecedent3 is “average”, then overall_rating is “medium.”
9. If antecedent4 is “average”, then overall_rating is “medium.”
10. If antecedent5 is “average”, then overall_rating is “medium” with .50.
11. If antecedent1 is “good”, then overall_rating is “high.”
12. If antecedent2 is “good”, then overall_rating is “high.”
13. If antecedent3 is “good”, then overall_rating is “high.”
14. If antecedent4 is “good”, then overall_rating is “high.”
15. If antecedent5 is “good”, then overall_rating is “high” with .50.

The “with .50” fragment means this rule is weighed half as much as it would be if that phrase is not there.

These rules are implemented in Python as shown in Fig. 5.

```
rule1 = ctrl.Rule(antecedent1['poor'], overall_rating['low'])
rule2 = ctrl.Rule(antecedent2['poor'], overall_rating['low'])
rule3 = ctrl.Rule(antecedent3['poor'], overall_rating['low'])
rule4 = ctrl.Rule(antecedent4['poor'], overall_rating['low'])
rule5 = ctrl.Rule(antecedent5['poor'], overall_rating['low'] % .50)

rule6 = ctrl.Rule(antecedent1['average'], overall_rating['medium'])
rule7 = ctrl.Rule(antecedent2['average'], overall_rating['medium'])
rule8 = ctrl.Rule(antecedent3['average'], overall_rating['medium'])
rule9 = ctrl.Rule(antecedent4['average'], overall_rating['medium'])
rule10 = ctrl.Rule(antecedent5['average'], overall_rating['medium'] % .50)

rule11 = ctrl.Rule(antecedent1['good'], overall_rating['high'])
rule12 = ctrl.Rule(antecedent2['good'], overall_rating['high'])
rule13 = ctrl.Rule(antecedent3['good'], overall_rating['high'])
rule14 = ctrl.Rule(antecedent4['good'], overall_rating['high'])
rule15 = ctrl.Rule(antecedent5['good'], overall_rating['high'] % .50)
```

Fig. 5. *Python implementation of fuzzy rules.*

After the user submits a form, the fuzzy calculator gets all five of the user's answers and relates them to each antecedent in the fuzzy control system. The degree of membership for each fuzzy set for each antecedent is calculated. The fuzzy rules are applied to determine the consequent to be defuzzified. The consequence is defuzzified using the "Center of Gravity" defuzzification method.

An example is presented. Suppose the user chooses the following values for each antecedent: 8 for antecedent1, 3 for antecedent2, 10 for antecedent 3, 9 for antecedent4, 1 for antecedent5. These values are correlated to values of membership for all three fuzzy sets in every antecedent. The membership values in each fuzzy rule for these inputs are given in Table 1.

TABLE 1. FUZZY RULES AND MEMBERSHIP VALUES IN OUR EXAMPLE.

Rule	Membership Value
1	0.0
2	0.4
3	0.4
4	0.4
5	0.4
6	0.4
7	0.6
8	0.6
9	0.6
10	0.6
11	0.6
12	0.6
13	1.0
14	1.0
15	1.0

The final values from the fuzzy rules are used to determine the graphical output for the consequent's overall_output. The final defuzzified result is calculated by finding the center of gravity of the graph generated by the fuzzy rules. This graph and its defuzzified output are represented by the vertical black line displayed in Fig. 6.

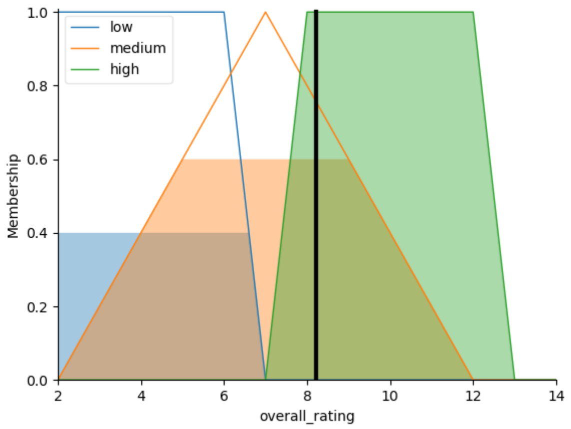
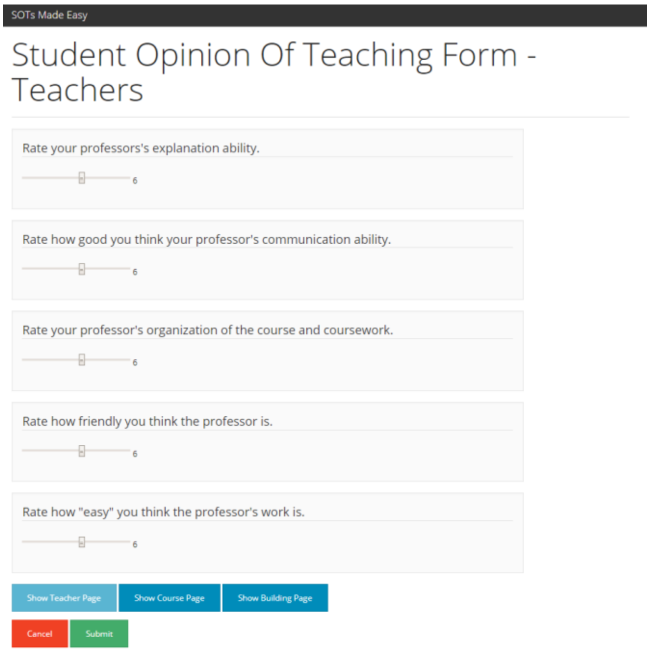


Fig. 6. *Graph of the overall_output with the defuzzified result.*

The defuzzified result is approximately 8.21. We then multiply by 10 and apply a percentage sign to make the answer 82.1%. This represents the final score for this professor, class, or building. Although this is not what is actually happening, this result can be thought of as a curbed score for this survey.

This graphical user interface of a webpage can be used by an actual university shown in Fig. 7. As with similar designs for web pages, a simple Cascading Style Sheets (CSS) is implemented with a straightforward layout. A variant from similar systems uses a direct slider versus radio circles for the selection.


SOTs Made Easy

Student Opinion Of Teaching Form - Teachers

Rate your professors's explanation ability.
6

Rate how good you think your professor's communication ability.
6

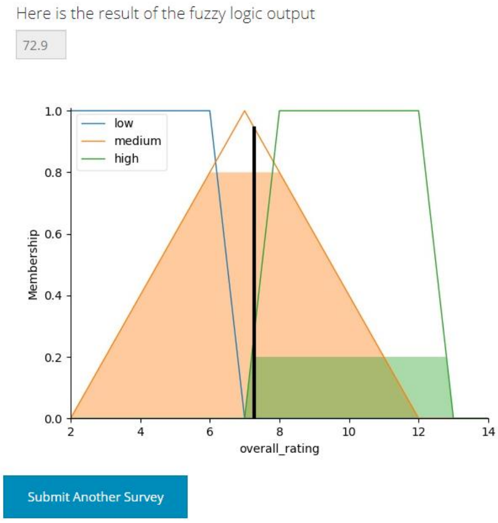
Rate your professor's organization of the course and coursework.
6

Rate how friendly you think the professor is.
6

Rate how "easy" you think the professor's work is.
6

Fig. 7. *The base layout of the webpage*

On the results page, we display the fuzzy logic graphs to illustrate why the logic generates its output as shown in Fig. 8. For the result, we show both the graph and the actual output as given by the fuzzy calculator.


Here is the result of the fuzzy logic output

72.9

Fig. 8. *The results of input from the webpage.*

IV. Discussion

This project is designed to relay information about learning to the reader, where we can compare our work to what is the standard. In our case, this would be a survey called “student opinion of teaching” or SOT. Since the SOT is a normal survey in which students usually rate a learning environment from 1 to 10, we test the results by comparing our solution to an expected result from an average survey. Our method produces results that are comparable with traditional methods. However, there are some extreme cases in which a given survey is different from what we will expect to see from an average survey. Even though we use a survey to compare all our results, there is more information to be gained from using our program.

Using fuzzy logic, we are able to give results that are more than just pure numbers. The results that are produced from our survey are accompanied by a relation to descriptive words such as bad, average, and good. For example, a score of 100 will be in the good rating and overall, we will rate this as “good”. However, a rating of 80 would be on the good side and on the average part of the graph making the rating “above average.” With these numerical ratings accompanied with an English descriptor, the person evaluating the performance of the learning environment has a much clearer picture.

What is interesting about our approach versus a weighted average of the results is our approach handles ambiguous information. We take advantage of this information that many students will offer slightly different answers to, but only in combination with the other answers. For example, it is very easy for a room to be aesthetically “average”, but difficult to be “good” or “bad”. On top of that, this particular result only really matters if the other results are either “good” or “bad” and tends to be irrelevant to most students when the other factors are average because most students are largely indifferent to room aesthetics. Using the fuzzy logic approach to solve the SOT problem, we can effectively curve factors dependently on one another versus independently as in a weighted average.

V. Conclusion

In conclusion, this project solution demonstrates fuzzy logic could reasonably be used to assist university leaders in their goal to provide quality education, specifically in their evaluations of university faculties, courses, and facilities. In the future, we plan to implement other features, such as a full complete database to store permanent results from different users.

References

[1] C. C. Lee, “Fuzzy Logic in Control Systems: Fuzzy Logic Controller. I,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 20, no. 2, pp. 404–418, DOI:10.1109/21.52551, March-April 1990.

[2] C. C. Lee, “Fuzzy Logic in Control Systems: Fuzzy Logic Controller. II,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 20, no. 2, pp. 419–435, DOI:10.1109/21.52551, March-April, 1990.

[3] J. A. Bernard, “Use of a Rule-Based System for Process Control,” IEEE Control Systems Magazine, vol. 8, no. 5, pp. 3–13, DOI:10.1109/37.7735, October 1988.

[4] T. Yamakawa, and T. Miki, “The Current Mode Fuzzy Logic Integrated Circuits Fabricated by the Standard CMOS Process,” IEEE Transactions on Computers, C-35, no. 2, pp. 161–167, DOI:10.1109/TC.1986.1676734, February 1986.

[5] C. Pereira, A. Tettamanzi, B. Liao, A. Malerba, A. Rotolo, and L. Torre, “Combining Fuzzy Logic and Formal Argumentation for Legal Interpretation,” Proceedings of the 16th edition of the International Conference on Artificial Intelligence and Law (ICAIL '17), pp. 49-58, DOI: https://doi.org/10.1145/3086512.3086532, June 2017.

[6] S. Petrudi, M. Pirouz and B. Pirouz, "Application of Fuzzy Logic for Performance Evaluation of Academic Students," 2013 13th Iranian Conference on Fuzzy Systems (IFSC), pp. 1-5, DOI: 10.1109/IFSC.2013.6675615, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6675615&isnumber=6675578, Qazvin, August 2013.

[7] Meenakshi and P. Nagar, "Fuzzy Logic Based Expert System for Students' Performance Evaluation," 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 803-808, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7100360&isnumber=7100186, New Delhi, March 2015.

[8] D. H. Rao, S. R. Mangalwede, and V. B. Deshmukh, "Student Performance Evaluation Model Based on Scoring Rubric Tool for Network Analysis Subject using Fuzzy Logic," 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), pp. 1-5, DOI: 10.1109/ICEECCOT.2017.8284623, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8284623&isnumber=8284511, Mysuru, December 2017.

[9] M. Tsiakmaki, G. Kostopoulos, S. Kotsiantis, and O. Ragos, “Fuzzy-based Active Learning for Predicting Student Academic Performance,” Proceedings of the 6th International Conference on Engineering & MIS 2020 (ICEMIS '20), no. 87,

pp. 1-6, DOI: 10.1145/3410352.3410823, September 2020.

[10] S. Ammar and R. Wright, "A Fuzzy Logic Aproach to Performance Evaluation," Proceedings of 3rd International Symposium on Uncertainty Modeling and Analysis and Annual Conference of the North American Fuzzy Information Processing Society, pp. 246-251, DOI: 10.1109/ISUMA.1995.527701 URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=527701&isnumber=11529, College Park, Marylang, 1995.

[11] R. Yadav, A. Soni, and S. Pal, “A Study of Academic Performance Evaluation Using Fuzzy Logic Techniques,” 2014 International Conference on Computing for Sustainable Global Development (INDIACom), pp. 48-53, DOI: 10.1109/IndiaCom.2014.6828010, March 2014.