

Improving usability and accessibility of Fuzzy Logic software systems with a web-based approach

Submitted MAY 2014 in partial fulfilment of the conditions of the award of
the degree MSci (Hons) Computer Science

Craig Knott
cxk01u

With Supervision from Jon Garibaldi

School of Computer Science and Information Technology
University of Nottingham

I hereby declare that this dissertation is all my own work, except as indicated
in the text:

Signature _____

Date ____/____/____



Abstract

This project was both a practical implementation of an online fuzzy logic software system, and a research piece into the optimal design of a fuzzy logic system, and how fuzzy logic can be best introduced to novices of the field. Specifically, I implemented an online system for the creation and evaluation of fuzzy logic sets and systems, using mostly JavaScript and JQuery, with an R backend to deal with the processing requirements of fuzzy logic. **In the end, I found out that... (using facts and figures)**

Contents

1	Further Work	1
1.1	Back end Interoperability	1
1.2	More Usability Improvements	1
1.3	Type-2 Fuzzy Logic Support	1
1.4	Customisations	2
A	User Evaluation Test Instructions	1

1 Further Work

Whilst the system produced can be considered a complete system, there are still definitely improvements that could be made, and functionality that could be added. In this section, the main extensions to the system have been detailed, along with their reasoning for not being in the initial release.

1.1 Back end Interoperability

The purpose of the system is to be a web-front end, to a back end that can perform fuzzy inference. This means that the system should not be reliant on back end that it currently uses (FuzzyToolkitUoN), and it could be interchanged with some other back end (potentially with greater functionality, or better performance). This is a result of the system being written almost entirely in JavaScript, making the majority of it's functionality a client-sided affair, and not relying on the back end. The only functionality that the back end truly provides, is the ability to save files to the user's machine, and to evaluate the fuzzy system created. This means that, with the implementation of a simple API layer between the front end and back end, the back end could easily be swapped out to any back end that the user would want. This would allow the user extra functionality within the system, making it much more useful for the expert user. This could include such things as different inference types (like TSK), different t-norms, and t-conorms for the inference methods, and different visualisation techniques. This would make the system much more useful, as it currently can only perform Mamdani inference, with a limited number of inference methods, due to the limited scope of the back end.

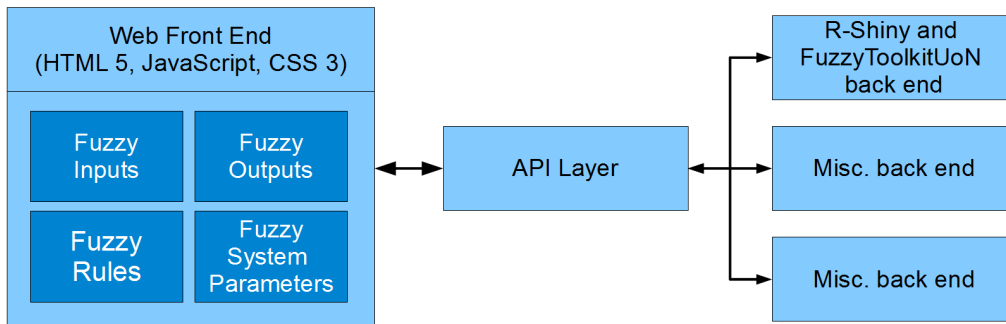


Figure 1: Interaction between the front end of the system and interchangeable back ends

1.2 More Usability Improvements

1.3 Type-2 Fuzzy Logic Support

The purpose of a type-1 fuzzy logic system, is to model uncertainty in the terms that humans use to describe properties. Unfortunately, this raises a paradox; fuzzy sets have a connotation of uncertainty, but the membership functions used are entirely certain once their parameters have been specified [3]. Type-2 fuzzy logic attempts to rectify this paradox, in the sense that uncertainty is not only limited to the terms used, but also the membership functions defined [1]. This means that each membership function defined has a third dimension, which allows an additional degree of freedom, to directly modal uncertainty [4].

As mentioned in the introduction to this report, type-2 fuzzy logic is not supported in this software. This is because the jump in knowledge between type-1 fuzzy logic (that *is* supported), and type-2, is extremely large, and would not be suitable for novice users. Other reasons include a lack of time to implement such a large feature, and that type-2

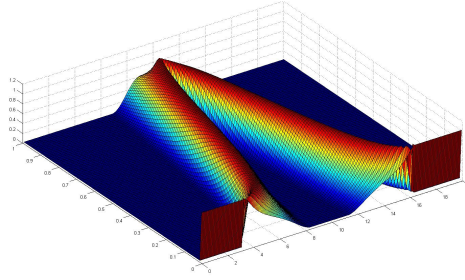


Figure 2: An example type-2 fuzzy set

fuzzy logic is still not widely adopted in the field. If more time was allocated to the project, this could have been implemented along side the type-1 system, however an entirely new back end would have been needed, as the current back end, FuzzyToolkitUoN, does not support type-2 fuzzy logic. Adding type-2 support to the system would have greatly improved it's usefulness and practicality, and could have easily made the tool a niche within the market, and the preferred software to use among experts. This would have been due to the ease of access, and ease of use of the system, which are features that most type-2 software systems currently available are lacking (such as Juzzy [5], which requires an installation of Java before it can be used).

1.4 Customisations

The ability to customise a website has been identified as one of the key factors for website success [2]. It has been found that the user's are more likely to return to a piece of software, if it is customised to their needs, and thus more personal. Unfortunately, there are not many customisations that could be made with this system, but some could be implemented to help improve user experience. One of these could be the look and feel of the system, which is currently a minimalistic white design. Some users prefer to use a darker interface (especially when completing work in the evening), and thus a set of different colour schemes could be employed, which would both serve a practical purpose, and an aesthetic one. Another potential customisation could be the ability to switch to an "expert" mode, where help is less present in the system, more keyboard short cuts are supported, and there is less GUI to navigate. This would greatly increase the speed in which an expert user could complete their task within the system, as there would be less distractions.

Words in text | some

Calculated with the TeXCount web service
<http://app.uio.no/ifi/textcount/online.php>

References

- [1] Oscar Castillo and Patricia Melin. Type-2 fuzzy logic. In *Soft Computing and Fractal Theory for Intelligent Manufacturing*, pages 33–46. Springer, 2003.
- [2] Wei-Shang Fan and Ming-Chun Tsai. Factors driving website success—the key role of internet customisation and the influence of website design quality and internet marketing strategy. *Total Quality Management*, 21(11):1141–1159, 2010.
- [3] Jerry M Mendel. Type-2 fuzzy sets: some questions and answers. *IEEE Neural Networks Society*, pages 10–13, 2003.
- [4] Jerry M Mendel and RI Bob John. Type-2 fuzzy sets made simple. *Fuzzy Systems, IEEE Transactions on*, 10(2):117–127, 2002.
- [5] Christian Wagner. Juzzy-a java based toolkit for type-2 fuzzy logic. In *Advances in Type-2 Fuzzy Logic Systems (T2FUZZ), 2013 IEEE Symposium on*, pages 45–52. IEEE, 2013.

A User Evaluation Test Instructions

The purpose of this test is to construct the fuzzy logic tipper example, in three different software systems, so that the strengths and weaknesses of each of these systems can be identified. The tipper example is a simple fuzzy system that uses food quality (rated from 0 to 10) and service quality (rated from 0 to 10) as inputs, to determine how much of a tip should be left (between 0 and 30 percent).

Please follow the instructions below, and do not hesitate to ask for help if you are stuck (this is not a test!)

1. test