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

Abstract giving a short overview of the work in your project

Why did you undertake the study? What were you examining, or testing or investigating. Return to your research question and ensure you have re-stated it concisely, coherently and clearly. A good opening is often, "The report examines . . . ".
What was done and how did you do it? Be specific, dont make generalised comments.

What did you find out? State specific outcomes and, if appropriate, draw conclusions. “The results found that 85% of respondents used non-standardised assessments”

Ensure to mention:

1. What is fuzzy
2. Further work with Type 2
3. Extend with other systems (Joe’s dissertation)
4. Changes that Luke makes
5. Friendly errors
6. Things from presentation
7. KeyPress Javascript library
8. Help system is dedicated, but offers links to other, helpful, external resources
9. only the evaluation step requires the internet (other than initial launch)

Contents

1 Introduction

Fuzzy logic is an ever expanding field, and as such, the tools we are using to work in this field should also be expanding. It is also important that the merits of fuzzy logic are made apparent to those other than the experts of this field, as this would help to produce more advanced control systems in the future.

Many software systems for working with fuzzy logic have already been produced, of which many different approaches have been attempted, and been successful to various degrees. Examples of such systems include: The MATLAB Fuzzy Toolbox[1], An R Package named FuzzyToolkitUoN[2], XFuzzy[3], and fuzzyTECH[4] (a more comphrehensive overview of these systems can be found in section 3.2).

These system are all worthwhile pieces of software, and they fulfil their main objective of allowing for the creation of fuzzy systems. However, whilst researching these systems as part of my second year group project at the University of Nottingham (and actually working on one, in the case of FuzzyToolkitUoN), I noticed that there were two key flaws that the majority of popular fuzzy software systems suffered from: difficulty of use, or difficulty of access (or even both).

The main objective of this project is to produce a software solution for the creation, manipulation, and inferencing of a fuzzy logic system, which is accessible online. With a specific focus on solving the issues that are faced by fuzzy logic software systems that are currently used (difficulty of access and use).

Many different techniques will be employed in solving these fundamental problems, to hopefully create a system that is as easy to use, and as easy to access as possible. Some of these techniques will include: online access; the ability to work with multiple file types, for cross compatibility; an intuitive design; unrestricted navigation, giving the user complete control and freedom; a dedicated, unobtrusive help system, to offer help to those that need it, but not to bother those that do not; and to built in a way that allows for future expansions and extensibility. A comprehensive list of all aspects of the software system can be found in section 4.

It could be argued that *another* fuzzy logic software system is not necessary, as it has been demonstrated that there are already many available systems. But, in contrast to this view, I feel that currently available software suffers from certain issues (those mentioned above), and this project aims to resolve these issues, and attempt to spread the influence of fuzzy logic to those other than experts in the field.

There will, however, be certain areas that this software system will *not* be focusing on, as I do not believe they are relevant to the question posed in this research. Namely, this project will not be focusing on higher levels of fuzzy logic; it will only be focused on type-1. This is because the leap in difficulty from type-1 to type-2 fuzzy logic is very large, and type-2 is simply not a concept that I feel is suitable or appropriate to introduce beginners to. More on this topic, including a definition of both terms, can be found in section 8.1.

[1] `http://www.mathworks.co.uk/products/fuzzy-logic/`

[2] `http://cran.r-project.org/web/packages/FuzzyToolkitUoN/index.html`

[3] `http://www2.imse-cnm.csic.es/Xfuzzy/`

[4] `http://www.fuzzytech.com/`

2 Motivation

The motivation of this project is a simple one: to produce a fuzzy logic software system that is easy to access, and easy to use, to help promote the wider adoption of fuzzy logic. The problem with systems currently available is that they suffer from one of the two following pitfalls: difficulty of use or difficulty of access. This means that novices can find it very difficult to get into the field, the software available does not facilitate productive use, and even experts can be held back by the software they are using. Some specific issues include: locating systems to use, complex installation processes, cost to the user, unintuitive user interface, or a requirement of (a considerable amount of) prior knowledge.

As part of my second year group project at the University of Nottingham, I worked on an R Package called FuzzyToolkitUoN. The goal of this system was to expand upon work completed by the Intelligent Modelling and Analysis group[5], to facilitate the use of fuzzy logic within the R programming language. Whilst working on this project, my group and I conducted a large amount of research into existing fuzzy logic software systems and it was during this research period that I began to notice the two keys flaws that I have mentioned before. Unfortunately, due to the nature of the R programming language, and the package we were producing, our project too fell into one of these pitfalls - difficulty of use. Personally, I was frustrated with this, and that is one of the reasons for the birth of this project - remedying past mistakes.

As I have mentioned, I also believe the greater adoption of fuzzy logic would be beneficial, as it adds a new level of reasoning that classical logic simply cannot. As such, another side goal of this project is to make a system that is as easy to use as possible, regardless of the skill of the user in both terms of knowledge of fuzzy logic, and of using computer software in general. I hope to produce a project that will be not only very easy to access and use, but also help novices to the field to learn about what they are doing, as well as why they are doing it, to help them gain a greater understand of the field of fuzzy logic.

The project detailed in this dissertation will aim to implement a fuzzy logic software system, in a novel format (online), and to specifically avoid the common pitfalls I have observed of other similar systems. Being online, the system is already on the right path to solving the difficult of access problem, as the users will be able to access the system from wherever they are, and on what ever platform (as it will not require any plugins, like Java, or Flash). This also means it is more accessible to the novice user, or the computer novice, as they need only navigate to a website to use the system; there is no complex download and installation process.

I will be using knowledge from the field of Human Computer Interaction Module to ensure that the system is a user-friendly, and easy to pick up as possible. Since taking the aforementioned module, I have been attempting to increase the usability of all the interfaces I design, as I believe that user interaction with a system is extremely important, and the way it is design has a huge effect (a great example is imagining a “push” door, with a handle; this simply causes confusion for the user of the door!) Simplicity is important in this design, because studies have shown that users lose more than 40% of their time to frustration, and that in most of these cases, the user ends up angry at themselves, angry at the computer, or feeling a sense of helplessness[2]; which is obviously not ideal for a system that is attempting to help the user learn.

[5] `http://ima.ac.uk/`

3 Background Information & Research

3.1 What is Fuzzy Logic?

Fuzzy logic is a "natural" way of expressing uncertain or qualitative information [1]. It is a form of logic that deals with approximate reasoning, as opposed to fixed, exact values, like those found in classical logic (where we may only have properties being true, or false). Instead of these strict truth values, fuzzy logic systems have a range of truth, between 0 and 1. This makes fuzzy logic much better for handling and sorting data, and is an excellent choice for many control system applications, due to the way it mimics human control logic. Lotfi Zadeh, who formalised fuzzy logic in 1965, states that the key advantages of fuzzy logic are that it allows us to make rational decisions in environments of imprecision, uncertainty, and partiality of truth, and to perform a wide variety of physical and mental tasks, without any measurements or computations [5].

In a classical set, the membership, $\mu_A(x)$ of x, of a set, A, in universe, X, is defined:

$$\mu_A(x) = \begin{cases} 1, & \text{iff x} \in \text{A} \\ 0, & \text{iff x} \notin \text{A} \end{cases}$$

That is, the element is either in the set, or not. In a fuzzy set, however, we have grades of membership, which are real numbers in the interval, $\mu_A(x) \in [0, 1]$. Every member of a set has a membership grade to that set, depicting how true the property represented by that set is, for the given member [4]. The traditional syntax for representing members of a fuzzy set is given below (although a full working knowledge of fuzzy logic theory is not necessary for this project).

$$A = \mu_A(x_1)/x_1 + ... + \mu_A(x_n)/x_n$$

The easiest way to observe the merits of fuzzy logic are to look at terms that we humans use in our everyday life, and attempt to map these are crisp functions. For instances, terms like "hot", "cold", "tall", "short", are all terms that we understand very well, and use often. However, if we were asked to give *exact* values for tallness, or shortness, we would not be able to. At what cut-off point would a person change from being considered short, to being considered tall? Fuzzy logic helps to alleviate these impossible choices, by having varying differing degrees of membership, to certain properties. The example in figure 1 shows this using three linguistic variables to describe the height of a person. Instead of at one point being either tall, short, or medium height, we, at all times, belong to all properties, to a differing degree.

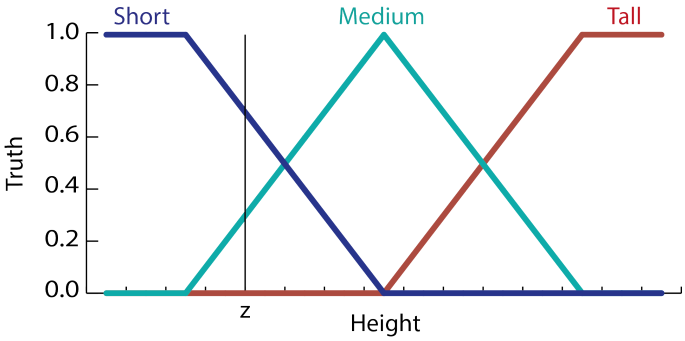

Figure 1: A fuzzy set depicting "height"

For instance, at the point labelled z, in the sets in figure 1, we belong in the "Short" set, to degree 0.7, we belong in the "Medium" set, to degree 0.3, and we belong in the "Tall" set, to degree 0.0. This is, naturally, much more precise than simply saying we are "Short", "Medium", or "Tall".

3.2 Existing Systems

Fuzzy logic has been around for almost 50 years now, and, with the rising age of the computer, it would be alarming if no software systems for its usage were in circulation. Luckily, this is not the case, and there are many examples of software systems focusing on the use of fuzzy logic, of which many different approaches have been attempted, to varying degrees of success. In this section, a number of these software systems will be evaluated, to discern their positive and negative qualities, to help improve the design of the project presented in this report.

MATLAB Fuzzy Toolbox
The first system to be explored is MATLAB's fuzzy toolbox, an add-on for the MATLAB software suite, to work with fuzzy sets and systems. This toolbox provides everything required to create type-1 fuzzy sets and systems, with relative ease. The main advantage it has over most other systems is that it has a graphical user interface, which makes a tasks like working with fuzzy sets (that require a lot of visualisation and updating in real time) much simpler. There is also an extensive library of documentation and tutorials available for both MATLAB, and this specific toolbox, that help novices to get acquainted with the system. These things both help to make the system very easy to use, and novice friendly.

Unfortunately, these positives do not outweigh the major disadvantage of MATLAB, and the fuzzy toolbox; which is that are pieces of proprietary software. This means that a novice to the field of fuzzy logic would have to invest a considerable sum of money, before they could even begin using the software. Whilst the system does have extensive documentation, and the user would be able to understand and use the system with relative ease, a piece of software does not require a large price tag to achieve this level of functionality and support. Another disadvantage of the MATLAB fuzzy toolbox is that is it not a dedicated piece of software, and is instead a limited subsection of the greater software of MATLAB. This means that the potential for extensibility is much less likely, as updates to the encompassing software would be deemed more important. It could even be argued that the installation of the MATLAB software, and then the installation of further software could be confusing to some novice users, which further alienates them.

FuzzyToolkitUoN
FuzzyToolkitUoN is an R-Package, produced by the Intelligent Modelling and Analysis group, at the University of Nottingham, that I personally worked on as part of my Second Year Group Project, at the aforementioned University. It provides functionality for the R Programming language to allow for work with fuzzy sets and fuzzy systems, including their evaluation. A major milestone for the project was it's official acceptance onto CRAN (The Comprehensive R Archive Network, an online library for R packages), in 2013. Being written in R, the package has access to the very powerful R processing tools, and graphics drawing capabilities, which help the user to visualise the system they are creating. Due to being hosted on CRAN, there is extensive documentation for every function in the package, including example usages, and explanations of all their parameters. This makes the usage of the package much simpler, because the user can easily access the documentation for any function they require, and there is a fully worked example for them to follow. Another advantage of the CRAN hosting is that any user with an R interpreter installed can access the package with only a few simple commands, helping the system to be much more accessible.

Unfortunately, there are downsides to working with the R language, the most prominent of which is the command line interface. Whilst the users have the features and functionality to plot the sets they are drawing, it can still be a cumbersome task, and does not promote ease of use. In a graphical user interface, the updating of graphs would be automatic, and the user could see their chances in real time, instead of having to change the graph, and then check what it looked like. Another issue with the package is that the descriptions of the functions, and the system itself, are specified so that a novice to fuzzy logic is not given the support they require; the system assumes that the user's knowledge of fuzzy logic is already somewhat sufficient.

XFuzzy (3.0)

Developed by the Institute of Microelectronics in Seville, Spain[6], XFuzzy (3.0), is a set of several tools, that cover the different stages of the creation of a fuzzy system. It allows for the construction of complex systems, whilst also offering flexibility, of allows the user to extend the set of available functions and tools. Each of the tools can be executed as an independent program, or, using the XFuzzy environment, can be integrated together as a graphical user interface, to ease the design process. As mentioned in the evaluation of the MATLAB fuzzy toolbox, a graphical user interface is a huge aid to the user in constructing a fuzzy system, as it allows them to see what changes they are making, and what effect these changes have. The software also runs on Java, which means it can be used on any operating system, as long as the Java Runtime Environment is installed, making it very easy to access.

There are, however, some issues with this software. These being that it is relatively unknown, is not updated frequently (the last update for version 3.0 was in 2012), and without using the graphical user interface, you are stuck using a command line interface, and must learn the system's bespoke language, in order to complete any tasks. The final disadvantage (which is also listed as an advantage, from a different view point), is the necessity for Java to be installed. This is an additional piece of software that this product is dependant on, and could further confuse the novice user with extra installations necessary.

fuzzyTECH

The final system that was observed was fuzzyTECH, another proprietary software, produced by INFORM[7]. It comes with a graphical user interface, making the interaction with the system much simpler than other software available. It also boasts that the application programmer does not require an understanding of fuzzy logic *or* programming to be able to use the system adequately - a feature that the software detailed in this report also hopes to boast. This claim is due to the extensive support functions provided with the software, which can be used to developer software that solves the problem at hand. Another advantage of fuzzyTECH is that is produces source code that can be used on various hardware platforms (for instance, PCs and micro-controllers).

The disadvantages of fuzzyTECH, just as with the MATLAB fuzzy toolbox, is that it is a pay-to-use piece of software. This greatly alienates the novice user, as they may not be willing to spend a large sum of money on a software they have never used, for a purpose they have never experienced before. There are also many different versions of fuzzyTECH, which would further confuse the novice user, as they may not necessary be aware of their specific needs, and thus be unsure as to which specific version would suit them best. In addition to this, fuzzyTECH has not been updated since early 2010, meaning the support that the novice user requires, may not be there - despite the systems extensive support functions.

[6] `http://www.imse-cnm.csic.es/`

[7] `http://www.inform-software.com/`

Talk about fuzzy toolkituon, matlab, xfuzzy (brief), fuzzytech(brief), mention their positions and negatives in terms of the key flaws, and novice users.

Then talk about my project and how this will counter the specific issues raised.

3.3 Platforms and Tools

1. Languages used (R, Javascript)
2. Web technologies (tools, languages)
3. Shiny r to html
4. Bootstrap
5. jquery
6. Good user interfaces

4 System Specification

Description of the work explaining what your project is meant to achieve, how it is meant to function, perhaps even a functional specification or user stories?

4.1 Functional Requirements

4.2 Non-Functional Requirements

5 System Designs

Design containing a comprehensive description of the design chosen, how it addresses the problem, and why it is designed the way it is make sure to include the diagram from the presentation

5.1 The Design Process

5.2 UI Design

5.2.1 First Iteration

“straight up javascript and html”, loses the power of R

5.2.2 Second Iteration

tabbed based system with long horizontal variables. maintaining large systems becomes more difficult. something about hci software science

5.2.3 Third Iteration

fully referenced explanation of why the current design is optimal

5.3 Navigation/Control Flow Design

5.4 Internal Design

George Miller- the magical number seven [3]. I didn’t want to bombard the user with too much information, hence the condensed view. (To help reduce to cognitive load on the user, the majority of screens have remnants of information from the previous screen on them (this is to combat the 7 ± 2 chunks of information in working memory))

6 Software Implementation

6.1 Key Implementation Decisions

6.2 a detailed description of all aspects of the system

6.3 Problems Encountered

Implementation containing a comprehensive description of the implementation of your software, including the language(s) and platform chosen, problems encountered, any changes made to the design as a result of the implementation, etc

7 Evaluation

7.1 Statistical Tests

7.2 User Feedback Tests

7.3 Successes and Limitations of the Project

Evaluation explaining how your software was tested (using different datasets or in different environments), statistical evaluation of performance, results of user evaluation questionnaires, etc.

8 Further Work

further... work?

8.1 Type-2 Fuzzy Logic

brief explanation of what it is, why it is different to type1, and why it was not included

8.2 Backend Interoperability

mostly in javascript, so only the inference engine needs to be ported

8.3 Customisations

nothing major, but small things to make people feel more at home if they are using the system a lot

9 Summary

Summary and further work including a personal reflection on your experience of the project and a critical appraisal of how the project went

Words in text	1

Calculated with the TeXCount web service
`http://app.uio.no/ifi/texcount/online.php`

References

[1] Pedro Albertos and Antonio Sala. Fuzzy logic controllers, advantages and drawbacks. IEEE transactions on control system technology, 1998.

[2] Jonathan Lazar, Adam Jones, and Ben Shneiderman. Workplace user frustration with computers: An exploratory investigation of the causes and severity. Behaviour & Information Technology, 25(03):239–251, 2006.

[3] George A Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychological review, 63(2):81, 1956.

[4] Lotfi A Zadeh. Fuzzy sets. Information and control, 8(3):338–353, 1965.

[5] Lotfi A Zadeh. From computing with numbers to computing with words. from manipulation of measurements to manipulation of perceptions. Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on, 46(1):105–119, 1999.

A test