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)

Ensure to mention:

1. Changes that Luke makes

2. Friendly errors

3. Things from presentation

4. KeyPress Javascript library

5. Help system is dedicated, but offers links to other, helpful, external resources

6. 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:

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

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.
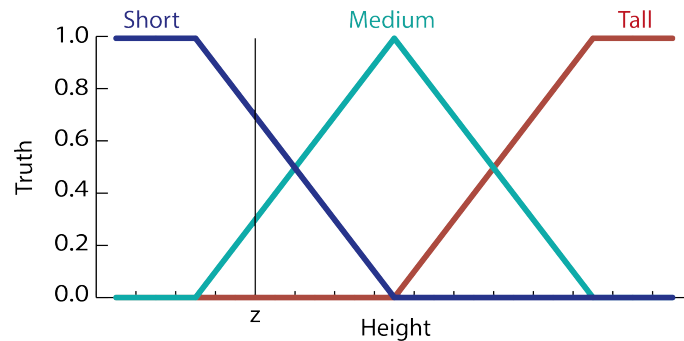

Figure 1: A fuzzy set depicting "height"

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/

3.3   Platforms and Tools

This section introduces some potential platforms and tools that could have been used in the project, along with justifications for and against. The system itself consisted of two major parts, the web front end, and potentially some server back end (at the research stage of the project it had not yet been decided whether the application was to be entirely client-side or not). Both front end tools (including design frameworks and programming languages) and back end tools were explored and evaluated during this process to discern which would be the most appropriate for this project. Further to this, there was some research into graphing tools to be used on the website, as drawing fuzzy sets as graphs is an extremely powerful and intuitive way to represent them. A full list of the final decisions of tools to use, including their justifications, can be found in section 6.1.

3.3.1   System Back End

Talk about mutliple ones for different languages, talk about shiny (can use a lot of a little). talk about fuzzy toolkituon

no database or such needed, just need a simple server to do calculations - could /really/ be client sided, but can't use R that way - optimal would be to link with fuzzytoolkituon (mean that updates to fuz could be applied to the website with reletive ease, if some api was created)

R-Node

R-Shiny

Node.js

3.3.2   Front End Programming Language

Naturally, the website would be built using HTML and CSS, as they are the de-facto standard languages for the construction and design of websites. However, the most appropriate language to provide the functionality to the website required some research. Fortunately, it was not necessary to decide upon only a single language for this task, as many web based programming languages combine with relative ease, and even complement one another.

JavaScript
JavaScript is an obvious choice when looking to provide functionality to a website, due to it's high level of integration with HTML. It is a client-sided programming language used to alter the content of a display document, or in this case, website. The main advantage of JavaScript is it's ability to directly and easily manipulate the content of a web page, by accessing the document object model. The basic semantics and syntax of JavaScript are very similar to many other languages, meaning adopting this language is very easy to both expert and novice programmers. But, it also provides advanced features and functionality, that help to produce extremely powerful systems with relative ease. The disadvantages of JavaScript are that there are security issues (although this is not an issue with the system proposed in this project), and there is potential that JavaScript can be rendered differently depending on layout engine, causing inconsistencies.

Flash

JQuery

3.3.3   Front End Design Framework

A front end framework, in the context of web development, is a standardised set of concepts, practices and criteria for dealing with the production of the front end of a website. There are many advantages to using pre-made front end frameworks for developing web applications, the most prominent of which are that: they generally look more aesthetically appealing than a single individual would be able to produce, they provide access to a large selection of easy to implement dynamic user interface elements, and many of them deal with browser cross-compatibility automatically. During the research period of this project, the following three candidates were evaluated: Semantic, Twitter's Bootstrap, and HTML KickStart.

Semantic
About semantic

Twitter's Bootstrap
Twitter's Bootstrap[8] is an extremely popular web development front end (so popular in fact, that some individuals[9] are beginning to complain about it's overuse), which helps to produce responsive websites, that function both on desktop and mobile devices. It is designed so that anyone of any skill level, beginner to expert, can pick up the framework and begin creating websites with relative ease. It provides a framework that produces websites that easily and efficiently scale for a multitude of devices, such as phones, tablets, and desktops, helping to make them much more accessible (although, not entirely relevant for this project, as there are many logistical issues with attempting to create a fuzzy logic system from a mobile device). There is also extensive and comprehensive documentation available for Bootstrap, including live working examples, and ready-to-use code samples, making its adoption into any project extremely simple. As mentioned above, however, Bootstrap is so popular nowadays that many people find it's usage off putting, as a large number of websites are beginning to look like clones of one another. For this reason, if Bootstrap were to be used in this project, it would be important to ensure that it was not one of these "sheep" websites.

HTML KickStart
About html kickstart

3.3.4   Graphing Tools

The drawing of fuzzy sets is extremely intuitive and powerful in a graphical format. For this reason, a suitable tool for the drawing of graphs on a website was required for this project. This would help to make the system considerably easier to use (as the user would be able to see their work as they were completing it), would take advantage of the graphical interface, and help to improve the general aesthetic of the system.

Directly from R

Google Charts

Highcharts

---

[8] http://getbootstrap.com/
[9] http://css.dzone.com/articles/please-stop-using-twitter

The first option for this was to the ability to draw graphs present in FuzzyToolkitUoN, which was one of the potential back end candidates. Using FuzzyToolkitUoN to draw the graphs would have promoted uniformity between the front end of the system, and it's back end, and would mean that the actual implementation of drawing of graphs would not need to be considered (as the system would simply be returning the image from R). It would also mean that the computation of the values for the graphs could be done entirely in R, and thus there would be less computation necessary on the client side. However, this presented the issue that transferring graphical data from an R back end to a web front is a difficult task, and the graphs themselves were simply pictures, and the graphs would not be interactive or dynamic.

Next, Google Charts[10] was evaluated. Google Charts is an entirely free tool that provides a variety of graph types, that are simple to generate and embed on a web page.

Finally, a product called HighCharts[11] was examined. This was a commercial piece of software, but could be used for free, if the project it was being used by was non-commercial (like this project).

Google charts, charts from R, highchart Js(free if not for profit!) (graphing tools to make it all easy)

Jquery for power extra features, javascript as main languageas tis integrates with html seemlessly

different web tools, jquery, html, css, javascript,

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 how the system will achieve it's goal, and how it is intended to function

4.1   Functional Requirements

A list of the functional requirements that the system must abide by

4.2   Non-Functional Requirements

A list of the non-functional requirements that the system must abide by

5   System Designs

Display and discuss all of the designs of the system, including the multiple iterations this went through. Mention how the designs address the problem, why they are designed the way they are, including software science, 13 sins of design, usability heuristics, and

---

[10]https://developers.google.com/chart/
[11]http://www.highcharts.com/products/highcharts

golden rules for 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))

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

Section to detail the navigation and flow of the system, and all design choices as a result of that

5.4   Internal Design

Show how the backend and foreground interact, and how this was designed (see picture from presentation)

6   Software Implementation

6.1   Key Implementation Decisions

A list of the tools and platforms that were used in the project, including justification for their inclusion. (To cover: Languages used, Web technologies used, Shiny, Bootstrap, JQuery)

6.2   Detailed Description of the Implemented System

A detailed description of the system, including how the individual sections were implemented, how they interact, strengths and weaknesses of individual components. Talk about the system as a whole as well, covering how the parts interacted and how well they did so.

6.3   Problems Encountered

A description of any major problems encountered during the implementation of the project, causes of these, and their resolutions - if there were any. Talk about the impact of the problems on the project as a whole, and how this potentially affected it (like changes to designs, etc)

7   Evaluation

7.1   Functionality Tests

Tests that look at the functionality of the system, including the individual components and the communication between the different parts. Give a list of tests, and the results of each of these tests, to ensure each functional requirement has been met.

7.2 User Feedback Tests

Large section detailing the user feedback tests as a result of surveys and demos with users using MATLAB, FuzzyToolkitUoN, and my new system. Draw conclusions from this data, and explain it thoroughly.

7.3 Successes and Limitations of the Project

Explain what went well with the project, what didn't go so well, and what could be done better for next time (with concrete goals).

8 Further Work

A list of things that could be done to expand the system in the future / things I did not have time to complete

8.1 Type-2 Fuzzy Logic

To include a briewf explanation of what type 2 fuzzy logic is, and why it is different to type 1. Also to include why it was not included.

8.2 Backend Interoperability

Talk about how the majority of the system is written in JavaScript, so that the system could be ported to different inference engine backends with little work.

8.3 Customisations

Talk about potential for a site that is more customisable for the user, as this would help them feel more at home, and make the system more effective for them. But also mention how basic customisations (like changing the background colour), wouldn't really make that much of a difference

9 Summary

This section will include a summary of the project as a whole, including my personal reflections on the experience of working on it, and a critical review of how the project went

Words in text │ 2,839

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