FuzziCalc: The Fuzzy Logic Spreadsheet*

Marek Sergot

Department of Computing,
Imperial College London, UK,
m.sergot@imperial.ac.uk

Abstract. This short note reports on joint work with Trevor Bench-Capon on FuzziCalc, an advanced, revolutionary, innovative fuzzy inference system whose social and commercial potential remains untapped. I summarise the design, theoretical foundations, and implementation, and sketch three illustrative applications. Some conclusions are drawn.

1 Introduction

In 1985 Trevor Bench-Capon and I wrote an article on possible approaches to the formal treatment of open texture in law (Bench-Capon and Sergot, 1988). The first part of the paper was an introduction to the concept of open texture and connections to the related concept of vagueness. (Professor Bench-Capon was responsible for most of that section.) The rest of the paper discussed three possible approaches. The first was approximation: the observation that for many practical purposes, especially in the everyday administration of law, a vague or open-textured concept could be and in practice often is substituted by an approximating crisp concept. (Professor Bench-Capon was responsible for most of that section.) The second approach was fuzzy logic. We speculated that sooner or later someone would suggest the application of fuzzy logic, either in Zadeh's original formulation (Zadeh, 1975) or in some modified form, to the treatment not only of vagueness but of open-texture in law. We wanted to pre-empt that suggestion and record several fundamental objections to it. (Professor Bench-Capon was responsible for most of that section.) The third approach, and the one we picked out as the most promising and deserving of further study, was a sketch of how open texture could be addressed through argumentation. (Professor Bench-Capon was responsible for most of the ideas in that section.)

Besides his fundamental contributions to the theory of argumentation and its applications, Trevor Bench-Capon has gone on to pursue the lines of research we had identified in the concluding part of our paper: to demonstrate how example-driven, case-based reasoning approaches to legal precedent can be reconciled and reconstructed in terms of argumentation, see e.g. (Bench-Capon and Sartor, 2003) *inter alia*, and to look at the evaluation of arguments and questions of what it is that makes an argument persuasive.

* A contribution to a *Festschrift* in Honour of Trevor Bench-Capon on the Occasion of his 60th Birthday.

My original idea for this *Festschrift* contribution was to follow up some of the ideas in his seminal paper (Bench-Capon, 2003) on value-based argumentation and in particular the very convincing examples used there to illustrate its application to the treatment of moral dilemmas. Subsequent developments in joint work with Katie Atkinson (Atkinson and Bench-Capon, 2006, 2007) produced a methodology for constructing value-based argumentation frameworks for practical reasoning. I thought I might look at how these methods address some of the classic problems in moral and ethical reasoning, or produce an implementation to automate the methodology, or both. However, that turned out to be a much larger undertaking than I had allowed for. The implementation in particular turns up a number of points of detail and further questions that deserve careful discussion and a much longer and detailed exposition than would be appropriate for this volume. I will save it for another occasion.

Still, I could not let this celebration pass without making some contribution. I have picked out another, less well known piece of joint work, the design and development of a prototype system that came to be known as FuzziCalc (The Fuzzy Logic Spreasheet). That work could not be published previously because of commercial embargoes and other constraints. It grew out of our discussions on the possible applications of fuzzy logic to aspects of law, and built on earlier ideas Trevor Bench-Capon had sketched out to me on the representation of vague concepts in rule-based expert systems. FuzziCalc was subsequently developed to industrial strength by Charlotte Anne Software[1]. The industrial versions will not be covered in this short note.

2 Design and implementation

The design and operation of FuzziCalc is perhaps easiest to explain by reference to a specific concrete example. The application we used to drive the development was a system designed to provide qualified advice about suggested careers based on a person's performance in school examinations.

Input to the system were the numerical marks obtained by the subject in school examinations. In the implementation, these appeared in a column of cells on the left hand side of the screen. We experimented with other arrangements, for instance, diagonal from top-right to bottom-left of the screen, but ergonomic studies identified the columnar arrangement as the most natural and appealing. Output from the system were the suggested careers for that person, qualified by truth value. These appeared in a column of cells on the right hand side of the screen, ordered by truth value, highest to lowest. A modification to an examination mark in an input data cell generated an immediate revision of the list of suggested careers, as in a conventional spreadsheet, except of course that in FuzziCalc's more sophisticated manifestation, all computations are qualified to indicate the truth of computed values.

In fuzzy logic as introduced in (Zadeh, 1975), the truth value assigned to a proposition such as 'the subject is suited to a career in journalism' is not

[1] Company motto: *lucrum per obfuscandum.*

restricted to the two values *true* and *false* but can be any value in the real interval between 0 and 1, with 0 corresponding to *false* and 1 to *true*. Thus, for example, if it is 0.693 true that Peter is suited to a career in journalism and 0.618 true that Jim is suited to a career in journalism, then not only is Peter more suited to a career in journalism than Jim, but we have a precise measure of how much truer the first statement is than the second. The value of having such precision at our disposal will not be lost on the reader. It is important to stress that fuzzy truth values are *not* to be confused with probabilities or likelihood measures of any kind. We are not concerned with the *probability* that Peter is suited to a career in journalism (for what could *that* possibly mean?) but the degree to which it is true that he is so suited, which is of course quite clear.

Knowledge about the characteristics and requirements of various career paths was encoded in the form of biconditional rules such as the following:

```
journalism ↔
    (English:good ∨ English:very-good) ∧
    Science:weak ∧
    Mathematics:weak ∧
    (History:good ∨ Geography:good)
```

The rule is for illustration only. The actual rules employed were more elaborate than that shown here. Also not shown is the concrete syntax provided in FuzziCalc to make the formulation of rules more concise. An intermediate concept such as `Science:weak` could be defined in similar fashion, for example (in simplified form, for the sake of illustration):

```
Science:weak ↔
    (Chemistry:weak ∧ Physics:weak)
```

Now, armed with truth values for the various factors appearing in the antecedents of rules (`English:good`, `Mathematics:weak`, and so on), it is a straightforward matter to calculate the truth value of the consequents of a rule (in the example, `journalism`) and of any intermediate concepts. In Zadeh's original formulation, the truth value of the conjunction $P \wedge Q$ is the minimum of the truth values of P and Q and the truth value of the disjunction $P \vee Q$ is the maximum of the truth values of P and Q. Many other variations have since been proposed; I will comment briefly below.

It remains to specify how the truth value of a factor such as `English:good` can be determined from the marks obtained in school examinations. This is done by means of a *truth profile*. A truth profile for a factor F is a mapping from (in the example, examination marks) to the real interval $[0, 1]$. Certain restrictions have to be imposed to ensure that truth profiles are sufficiently well behaved but these are technical details that need not detain us.

Example: examination marks of 30, 40, 50, 60, 70, 80 for English might map to truth values for `English:good` of (for illustration) 0.1, 0.2, 0.35, 0.9, 0.75, 0.6, respectively, to truth values for `English:very-good` of (say) 0, 0, 0.1, 0.2, 0.8, 0.95, to truth values for `English:weak` of (say) 1, 0.85, 0.3, 0.05, 0, 0, respectively, and so on.

Truth profiles Clearly, determining the truth profile for a given factor is no trivial matter. We experimented with many different shapes: Gaussians, sinusoidal curves of various kinds, more or less angular wave forms, and many others.

Correct choice of truth profile was determined through a rigorous process of structured disciplined debate, supported by extensive scholarship, and diagrams drawn using different coloured pens. The use of different coloured pens is important. It increases confidence and helps to ensure accuracy. It would obviously be a nonsense if we were to compute the truth value of `English:very-good` as 0.709 (say) when the actual truth value was only 0.686. It is essential to get these details right.

Much of our investigations throughout the development phases centred on establishing the correct shape for truth. Towards the end of that period, we came across a scholarly work in which the author argued by reference to historical sources[2] that truth is in fact *spherical* (in the material sense). Further exploration led to the development of a more advanced FuzziCalc engine employing ellipsoidal truth values instead of simple real numbers and a revolutionary three-dimensional spreadsheet as interface[3]. The details cannot be presented here for reasons I am not at liberty to disclose, or even refer to.

Technical problem 1 A technical problem that emerged early in the first prototype was that in many cases the system was unable to compute any output for a given set of input data. Worse, this also had the unsettling effect that the modification of an examination mark on the left hand side of the screen could often result in the complete disappearance from the screen of career possibilities on the right. The problem can be traced to the fact that there are many persons whose abilities, as measured by examination performance in school subjects, make them unsuited to any particular career. The knowledge base we had constructed merely reflected that fact. The problem remained unsolved until a flash of inspiration by Professor Bench-Capon illuminated the way forward. His insight was to observe that in practice persons unsuited to any particular career were ideally suited to a career in the Civil Service[4]. This remarkable insight led to the incorporation of a default conclusion (in the application, 'Civil Service') with truth value 1 to be triggered in the absence of any other computed value. The required modifications to the implementation of the engine and its theoretical underpinnings are immediate. I omit the details since they are easily reconstructed.

Technical problem 2 Efficiency of an inference engine is often an obstacle to its effective deployment. The problem in FuzziCalc was that the inference engine was *too fast*, which undermined the user's credulity that anything of significance

[2] Parmenides (5th century B.C.) though the reference is almost certainly spurious.

[3] Alternatively: a two-dimensional spreadsheet with three columns.

[4] This was true in the United Kingdom in the 1980s. We did not investigate whether it is a universal phenomenon. In modern times 'University administration' is probably the most widely applicable analogue.

was being computed. We were faced with two seemingly irreconcilable requirements: users need to be persuaded that an inference engine is labouring over its computations on the one hand, but demand an instant response to changes in input data values on the other. The problem was eventually solved by developing a compilation technique that transformed the FuzziCalc knowledge base into an efficient internal form when loaded by the user (or in modern terminology, when the 'app was opened'). The required transformation is so trivial however that we merely performed a series of heavy but completely pointless numerical calculations in order to give the illusion that something significant was happening. At that point the FuzziCalc engine was ready to accept user input (it had been 'unleashed' in the technical jargon).

Evaluation This was a fully functional proof-of-concept prototype. (In contrast to the references one sometimes sees in the literature to 'non-functional' prototypes, by which is meant usually 'non-functioning'.) No claims were made for the accuracy of the knowledge base. We took no responsibility for regrets and disappointments resulting from making life choices on the basis of FuzziCalc computations.

3 Two applications

FuzziCalc was applied to a range of important and substantial applications, of which I will sketch briefly just two.

3.1 Destination Advice for Travellers (*DAFT*)

DAFT was a system which offered advice to travellers seeking the perfect holiday, based on earlier preliminary work by Trevor Bench-Capon. The knowledge base specified in FuzziCalc rules the features and characteristics of popular holiday destinations: good beaches, sunshine, historical sites, culture, night life, facilities for children, cheap wine, availability of authentic English cuisine, and so on. Input to the system, on the left hand side of the screen, was a list of all such factors together with a slider for each which the user moved to record his or her own subjective assessment of the importance of that factor, in a scale from -10 (absence essential) to 10 (presence essential) with 0 indicating indifference. Output, on the right hand side of the screen, was an ordered list of suggested holiday destinations, qualified by truth value, naturally. Details of the knowledge base are proprietary and cannot be revealed. Truth profiles for each of the subjective factors were constructed following the methodology described above.

Evaluation: empirical, qualitative. *DAFT* was demonstrated at a series of international workshops and conferences, either as part of the industrial exhibition or informally during lunch and coffee breaks. Delegates were invited to try out the system for themselves, FuzziCalc's immediately intuitive interface making further detailed instruction unnecessary. User responses were carefully recorded.

They divided roughly into two equal parts: those who responded 'Do you know, I have always wanted to go there!' and those who replied 'That is amazing. I went there once. It was the best holiday I ever had!'

Comment: This unanimously enthusiastic response was most gratifying, yet it also raised an important research question. Everyone knows that Morocco, say, enjoys sunshine and that Rome has a rich collection of historical sites but besides commonplace information of that sort, the rules in the *DAFT* knowledge base were compiled *without knowing anything about the destinations covered* and *nothing* about the features we claimed they possessed. It is remarkable that the system was nevertheless able to generate detailed advice that, without exception, was perceived as valuable and accurate by its users. The explanation of how this could happen remains to be investigated.

3.2 Case law and precedent (*CLAP* and *CLAP-dash*)

The *CLAP* system and its subsequent refinement *CLAP-dash* were built to evaluate the potential applications of FuzziCalc to the construction of case law databases and the search for matching precedents. The *CLAP* knowledge base consisted of FuzziCalc rules whose consequent was a reference to a previously decided legal case and whose antecedent listed the factors deemed to have been influential in deciding the case, qualified by fuzzy terms such as *important*, *very-important*, *minor*, *negligible*, and so on. User input, as in *DAFT*, was the list of all potential influencing factors with sliders for recording the estimated degree of relevance of that factor to the new case to be analysed. Output from the system was an ordered list of previous cases, with a precise measure of how true it is that their decision applies to the circumstances of the new case.

For *CLAP*, accurate determination of the correct truth profiles turned out to be extremely problematic, which no amount of rigorous debate or coloured pens could resolve. Sometimes a smooth sinusoidal curve worked well; in other application domains a sharp saw-toothed shape for truth was clearly right. The extended version *CLAP-dash* provided therefore a wide range of truth profiles, giving the user the option of choosing between them *at run-time*. A strong argument was also put forward that for application to legal precedent, Zadeh's original truth tables had dubious theoretical basis, and that other variations modelled legal reasoning more accurately. The user was also given the option of choosing at run-time between different fuzzy logic calculi.

Evaluation: anecdotal. The *CLAP-dash* was demonstrated at an important international conference to an audience of lawyers and legal practitioners. The response was unanimously enthusiastic: 'Why, this is exactly the tool we have been waiting for!' was typical. In order to make the evaluation more systematic, each member of the audience was asked to rate the system. There were two choices: 'excellent' and 'superb'. Rankings divided roughly equally. (One assessor added a hand-written comment asking whether 'excellent' was better than 'superb' or 'superb' was better than 'excellent'. His or her ranking was discarded.)

For all its technical sophistication, the *CLAP-dash* experiment must be judged a failure. The knowledge base of cases had been deliberately constructed to demonstrate that the output depended critically on choice of truth profile and choice of fuzzy logic. Thus, if the shape of truth profile was changed from, say, sinusoidal to saw-tooth, a completely different set of legal precedents came into view in the right hand column. Disconcertingly, this did not dampen in any way the enthusiasm of the evaluation group. We call this 'the *DAFT* phenomenon'.

4 Conclusion

This account is of dubious historical and technical accuracy, though it is probably no worse in this respect than many of the published papers in the literature on applications of logic to representation problems in computer science. The truth of that last remark I would estimate at 0.887, or possibly a little higher.

FuzziCalc is an inference system whose foundations are built on shifting sand and whose conclusions ought therefore to be taken *cum grano salis*—a satisfyingly littoral metaphor on which to close an article dedicated to a person born in a naval establishment who has chosen to spend nearly all of his professional life in one of the world's great industrial ports.

This note reports on joint work with Trevor Bench-Capon. I am however solely responsible for any mistakes or misunderstandings, deliberate or otherwise. Technical details of FuzziCalc are by kind permission of Charlotte Anne Software.

Bibliography

K. Atkinson and T. Bench-Capon. Addressing moral problems through practical reasoning. In Lou Goble and Ch. Meyer, John-Jules, editors, *Deontic Logic and Artificial Normative Systems*, volume 4048 of *Lecture Notes in Computer Science*, pages 8–23. Springer, 2006.

K. Atkinson and T. Bench-Capon. Practical reasoning as presumptive argumentation using action based alternating transition systems. *Artificial Intelligence*, 171(10):855–874, 2007.

T. J. M. Bench-Capon and M. J. Sergot. Towards a rule-based representation of open texture in law. In Charles Walter, editor, *Computing Power and Legal Language*, pages 39–60. Greenwood/Quorum Press, Westport, 1988.

T.J.M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.

Trevor Bench-Capon and Giovanni Sartor. A model of legal reasoning with cases incorporating theories and values. *Artificial Intelligence*, 150(1):97–143, 2003.

L. Zadeh. Fuzzy logic and approximate reasoning. *Synthese*, 30:407–428, 1975.